BIT-LET

ötlet SZÁMÍTÁSTECHNIKAI MELLÉKLET 1986 MÁRCIUS 30



Lehet, hogy igazán világszínvonalú szoftverek már készülnek Magyarországon, magyar programozók talán tudnak ilyeneket produkálni. De hogy a programok magyarázatában, kezelési útmutatóinak készítésében még nem érjük el a világszínvonalat, annyi szent. Mert mit kellene tudnia egy ilyen leírásnak? Mindössze annyit, hogy a programot fölhasználó, programozáshoz nem értő is kitűnően értse az abban leírtakat, segítségével minden gond nélkül tudja használni a programot.

Megveszi a fölhasználó az ígéret szerint csecsemőnek is egyszerűen megtanulható fölhasználói szoftvert. Kicsit ugyan összehúzza a szemöldökét, amikor meglátja a kötetnyi használati utasítást, no de sebaj, gondolja, bizonyára egy csomó olyasmi is van benne, amire neki, egyszerű laikus fölhasználónak nincs szüksége, de a szakembereknek jó, ha leírták. Hazaviszi a programot, s kinyitja a magyarázó kötetet. Hálistennek az első oldal elmagyarázza neki, hogy kell betenni a lemezt a gépbe, meg hogy hogy kell betölteni azt. Idáig minden stimmel, ezt meg is értette, meg is csinálta. Most várja, hogy a könyvben majd ugyanaz az ábra következik a megfelelő magyarázattal, ami a képernyőjén megjelenik... Sajnos nem. A könyv szerzői előbb egy szép hosszú bevezetőt írtak, amelyből a fölhasználó megtudja, hogy mire is használhatja a programját. Persze ezt nagyjából már tudja, hiszen ezért vette meg a programot. Őt most az érdekelné, hogy mit kell megnyomnia, ha dolgozni akar a programmal. Ezt azonban nehezen tudja csak kisilabizálni a könyvből, mire megtalálja a megfelelő képernyőábrát, a hozzá tartozó magyarázattal, eltelik vagy félóra. Közben persze hatszáz más dologba kénytelen beleolvasni, ami nem árt, de nem is használ, hiszen úgysem érti, mert még nem tart ott. Ja, hogy egy ilyen felhasználói kézikönyvben kell lennie egy megfelelő mutatónak is, amelyből megtudhatja, hogy mit hol talál? Hát igen, ez a mutató olyan, amilyen, kicsit ellentmond a nagy számítógépes logikának.

Még folytathatnánk a történetet, részletezhetnénk, hogy hogyan adja föl a tanulást fölhasználó először, másodszor, azután hogyan kéri meg a programot már ismerő barátját, hogy tanítsa meg őt annak használatára, s hogyan döbben meg, hogy a barát milyen egyszerűen vezeti be a program rejtelmeibe. Írhatunk még arról, hogy a sikeres tanulás után hogyan jön rá a fölhasználó, hogy tulajdonképpen minden a képernyőre van írva, minden használati utasítás ott van, csak éppen bikkfanyelven fogalmazták, így hát készítenie kell egy szótárt magának, hogy tudja, mi mit jelent. A számítógépről azt állítjuk, írjuk, hogy kezelése gyerekjáték, bárki megtanulhatja, épeszű ember meg is kedveli, s pillanatok alatt megtalálja azt a tevékenységet, amelyet elvégezve, életét könnyítheti meg a gép. Ezt állítjuk, ám gyakran elfelejtjük, hogy ezt nem elég állítani, ahhoz, hogy ezt azok a bizonyos épeszűek is belássák, megfelelően egyszerűen kezelhető programokat kell a kezükbe adni, s még ennél is egyszerűbb magyarázatokat kell hozzá írni.

A legkorszerűbb gépek – Lysa, Macintosh, Amiga – már teljesen új alapokra helyezték ezt az egészet. Hálistennek az úgynevezett ablakos szoftverfilozófia lényege épp ezt a laikus fölhasználót tiszteli meg azzal, hogy a legegyszerűbb paraszti ész logikájával vezeti be őt a legbonyolultabb szoftverbe is. Egyelőre nálunk aligha várható e kint is drága gépek tömeges elterjedése. Így hát kénytelenek vagyunk egyelőre a hagyományos módon magyarázni a szoftvereket. Célunk, hogy minél több embert győzzünk meg a gépek hasznáról, mind több vezetőt vegyünk rá, hogy belássa, a gép vásárlása létérdek. Márpedig ezek a célok programozói bikkfanyelven írott magyarázatokkal, tudóskodóra sikeredett, néhol emészthetetlenül unalmas és olykor száraz TV BASIC sorozatokkal nem fognak megvalósulni. (Utóbbi ismétlését egy sokkal populárisabb sorozat elkészítése helyett – csak félmegoldásnak tartjuk.)

A képlet egyszerű. Egy tévé, egy lemezjátszó, egy magnó vásárlásakor elvárjuk, hogy ahhoz olyan használati utasítást kapjunk, amely a háziasszony számára is három perc alatt érthetővé teszi a gép használatát. Ugyanezt kell elvárni a gépek, szoftverek kezelési utasításától is! *Angyalosi László*

BELÜLRŐL

HÍROLDAL

Fonodában

A dunaújvárosi fésűsfonodában a Computext és a Budapesti Műszaki Egyetem által közösen kifejlesztett számítógépes rendszert helyeztek üzembe. Az adatgyűjtő és termelésirányító rendszer a keretorsózóról származó adatokat elemzi és az optimális sebességet vezérli. Alkalmazásával mintegy 10–15 százalékkal nőtt a termelékenység, miközben jelentősen csökken a hulladék mennyisége.

Paci chip

Japán kutatók megdöbbentő eredményt értek el a fehérje molekulák félvezető tulajdonságainak tanulmányozásában, illetve gyakorlati felhasználásában. Lovak szívéből kivont fehérjékből sikerült működő „paci" vagy „biochipet" előállítani, amely a jövőben akár számítógép építésére is alkalmas lehet. Kvantumkémiai vizsgálatok már két évtizeddel ezelőtt bizonyították, hogy a biológiai óriás molekulák félvezető tulajdonságokkal bírhatnak. De gyakorlati eredmény csak most született.

GYORS

A hagyományos folyadékkristályos kijelzőknél sokkal gyorsabban kapcsoló folyadékkristályt fejlesztettek ki a Központi Fizikai Kutató Intézet fejlesztő szakemberei. Az egyezred másodperc alatti kapcsolási sebesség lehetővé teszi, hogy a folyadékkristályt nagyméretű, kisfogyasztású speciális tv-képernyők gyártására is alkalmazhassák. Tekintve, hogy a nagy világcégek is erőfeszítéseket tesznek az ilyen folyadékkristályok kifejlesztésére komolyabb eredmények nélkül, a magyar siker üzleti szempontból is kihasználható lehet.

Könyvek között

Hogy kerül a csizma az asztalra, kérdezhetnénk a hír hallatán. Az ötlet azonban nem is rossz. Az Állami Könyvterjesztő Vállalat és a Novotrade Rt. megállapodása alapján öt vidéki város könyvesboltjaiban árusít a Művelt Nép számítógép programokat és kiegészítő eszközöket Commodore 16 és 64, valamint ZX Spectrum személyi számítógépekhez.

PC-piac!

A személyi számítógépek eladási száma a francia piacon az utóbbi években megkétszereződik. A dinamikus fejlődés alapja, hogy ma még a francia háztartásoknak alig több mint 2,5%-ában van személyi számítógép, ami tizedrésze például az angliai gépszámnak. Ugyanakkor a kisvállalatok is jó piaci területnek számítanak, hiszen alig 40%-uk használ pillanatnyilag számítógépet. Sőt, jó felvevő piac a francia iskolák, oktatási intézmények alkotta oktatásügyi ágazat is.

SDHT

Az amerikai Bell laboratóriumban kifejlesztették a világ leggyorsabb félvezető eszközét az SDHT tranzisztort. A gallium-arzenid technológiával létrehozott új eszközt elsősorban a kis bonyolultságú integrált áramkörök gyártásához, illetve a nagysebességű számítógépek és más elektrooptikai berendezések logikai áramköreiben használják majd.

Nagy RAM-ok

Alig csodálkozott el a világ a 256 kbit-es RAM-ok piaci megjelenésén és – az előirányzottnál évekkel előbb – máris bejelentették japán szakemberek az 1 Mbit-es RAM-ok piacrahozatalát. A japán Mitsubishi cég elsőként jelenteti meg az 1 Mbit-es olvasóerősítővel kombinált dinamikus RAM-ot.

MEGADOC

Új módszerét dolgozták ki a hollandiai Philips cég fejlesztői a dokumentumok gyors tárolásának és könnyű visszakeresésének. Megadoc elnevezésű berendezésükkel húsz A/4 oldalas dokumentumot oldalanként négymillió képpontra bontanak egy perc alatt. Ez a képfelbontás sokkal jobb a tv képfelbontás minőségénél. Az így létrejött elektronikus információ optikai lemezre kerül. De mielőtt tárolnák a nagy volumenű információtömeget csökkenteni, azaz mintegy összenyomni szükséges. Ez azt jelenti, hogy az 1–1 részletben azonos képi információt hordozó jeleket (pl. egybefüggő sötét részlet) nem kell egytől-egyik tárolni, hanem egy jelzőszámot, amely közli, hogy ott abból az értékű jelből hány van egymás mellett. Így előfordulhat pl., hogy egy A/4-es oldal négymillió jeléből mindössze százezer jel tárolására van szükség, és átlagosan egy A/4-es oldalból mintegy 50–60 ezer fér el egy harminc centiméter átmérőjű optikai lemezen.

KAZAH-HÍR

Kazah geológusok olyan számítógép-rendszert dolgoztak ki, amelybe betáplálva a lelőhelyre vonatkozó adatokat megkaphatják az érckészletekre vonatkozó információkat. Pontos képet szerezhetnek az érc eloszlására, minőségére, a lelőhely térbeli elhelyezkedésére, a kiaknázás lehetőségeire vonatkozóan. A rendszert több konkrét feladaton kipróbálták és igen jó eredményeket értek el.

Fényszámítógép

Japánban és az Egyesült Államokban egyaránt nagy ütemben kutatják a fényszámítógépek megalkotásának lehetőségeit. A fényszámítógépekben, ahol az információt fotonok továbbítják, az alapfolyamatok sebessége mintegy 250 ezerszerese a jelenlegi szupergépekének és az átvitel sebessége mintegy 10 Gbit/s lesz. A jelenlegi fejlesztésekben az egyes integrált áramkörök közötti jelforgalom biztosításához a holográfiát is igénybe veszik.

„Szemezés"

A számítógépekhez használatos képernyős terminálok egyre gyakrabban okoznak különféle szempanaszokat. Az esetek számának növekedése természetesen a technika általánossá válásával magyarázható. A betegek legtöbbször gyengén látásra, égető fájdalomra, könnyezésre, gyors szemkifáradásra panaszkodnak, melynek oka elsősorban az

asztigmatizmus okozta helytelen fénytörés és a helyiségek nem kielégítő megvilágítása. Megállapították, hogy ezeken kívül panaszt okoz az, hogy a terminál kezelőinek sokat kell a képernyőre és az adatokat, szöveget tartalmazó papírlapokra nézniük, és a változó távolság miatt a pupilla hosszú időn át folyamatosan kitágul és összehúzódik.
Jelentős befolyásoló tényező a képernyők színe, fényereje is.

Lízing

Az Általános Vállalkozási Bank RT, a Metrimpex Külkereskedelmi Vállalat és a SZÁMALK szerződése alapján a hazai számítástechnikai kultúra elterjesztésében meghatározó jelentőségű lépésre került sor: a SZÁMALK megkezdte a szocialista import számítástechnikai rendszerek komplett lízingjét. A jelenlegi tőkeszegény gazdasági környezetben a több éves részletekben, költségből fedezhető számítógépbérlet nagyban elősegítheti a vállalatok gyors számítógépesítését. A bérletezett rendszereket a SZÁMALK üzembe helyezi és megfelelő alapszoftvert, oktatást, egyéves garanciát, alkatrészt biztosít.

Prognózis!

Néhány éven belül kiépül hazánkban a meteorológiai műholdvevő és számítógépes feldolgozó rendszer. A rendszer képfeldolgozó számítógépe már rendelkezésre áll. Az év végéig a műholdvevők nagy részét is felszerelik. A műholdas, számítógépes hálózat segítségével a havi és kéthetes, valamint a napos előrejelzések mellett a legkeresettebb néhány órás területi időjárás-előrejelzésekre is lehetőség lesz.

Zöldség

A hír nem zöldség, vagyis komoly, hogy egy tokiói áruházban kísérleti salátagyárat működtetnek. A saláták gyors növekedését nátriumlámpák fényével és a levegőnél ötször nagyobb széndioxid-koncentrációval érik el. A vízben fejlődő saláták műtrágyázásának, fényigényének, hő- és párafelvételének mértékét egy mikroszámítógép vezérli.

A főváros vize!

A fővárosi ivóvízellátásnak előreláthatóan nemigen lesznek mennyiségi korlátai. Gondot okoz viszont a víz minősége. Budapest rendelkezik Európa legnagyobb folyóparti kútrendszerével és ha növekedne is a fogyasztás, azt a kúthálózat bővítésével ki lehetne elégíteni. Viszont a minőséget károsan befolyásolja a magas nitrát- és vastartalom. A legfontosabb megoldandó feladat a minőség javítása és emellett ki kell építeni a főváros vízrendszerének korszerű, számítógéppel vezérelt energia- és távközlőhálózatát.

Mátka

A 3000–5000 hektárnál nem nagyobb, hagyományos növénytermesztéssel és állattartással foglalkozó termelőszövetkezetek és állami gazdaságok céltudatos, hatékony költséggazdálkodását segíti elő a Számítástechnika-alkalmazási Vállalatnál (SZÁMALK) készülő MÁTKA mezőgazdasági számítógépes programcsomag.
A MÁTKA programcsomag több önállóan is alkalmazható alrendszerből áll. Jelenleg elkészült a „Gépgazdálkodás" elnevezésű, és most készülnek a „Növénytermesztés" és az „Állattartás" alrendszerek. A SZÁMALK szakemberei tervezik további – például „Élelmiszerfeldolgozás", „Háztáji gazdaságok szervezése" – alrendszerek kidolgozását is.

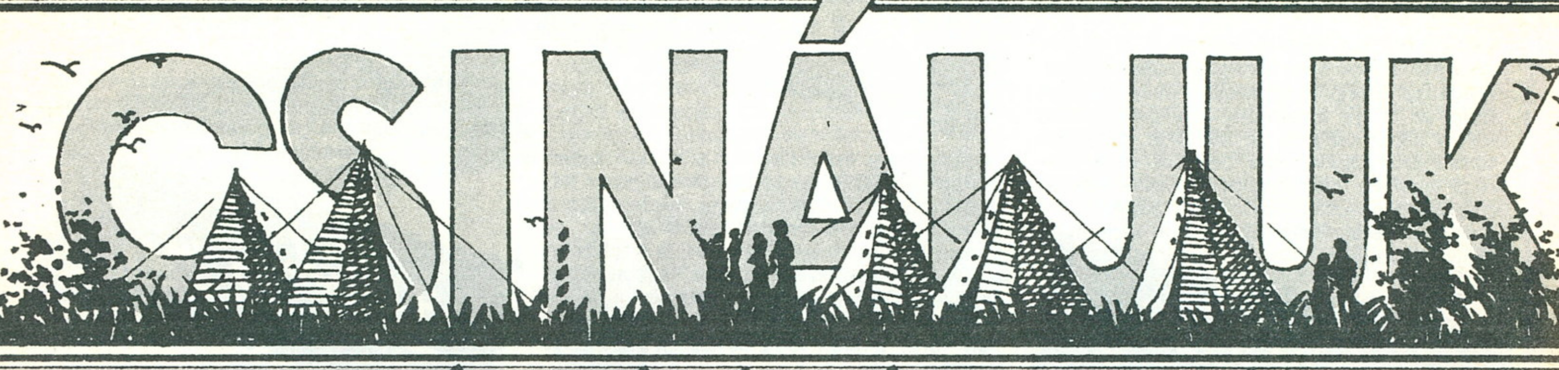


ÚJ!

A Panasonic cég új, SR. PARTNER elnevezésű személyi számítógépe nem igényli semmiféle periféria csatlakoztatását, mivel azokat beépítették: egy 80/132 hasábméretű nyomtatót, egy 9 inches nagyfelbontású képernyőt, egy 360 K-s lemezmeghajtót. Az IBM hardver- és szoftverkompatibilis Sr. Partneren futtathatók a legnépszerűbb üzleti programok. A Panasonic különösen ajánlja a WordStar, Visi Calc, MS–DOS 2.11 és a GW BASIC használatát. Beépített 256 K memóriája 512 K-ig bővíthető. A képernyő színes és grafikus üzemmódban is működtethető.

ötlet

MI HOGY CSINÁLJUK -a számítógépes tábort



Tavasz a nyári számítógépes táborok meghirdetésének, tervezésének, szervezésének ideje. Már többször írtunk lapunkban arról, hogy milyen fontos feladatnak tartanánk, hogy ezeknek a nyáriasított oktatási formáknak is legyen valamiféle szakmai, módszertani tapasztalatcseréje. E célunk megvalósítása érdekében közöljük Török Turul írását a KFKI táborainak szakmai tapasztalatairól, s már most jelezzük, hogy a kézirat nyomdába adásával egyidejűleg elküldtük annak másolatát néhány, táborokkal szintén foglalkozó pedagógusnak, szakembernek – azzal a kéréssel, írja meg saját elgondolásait, tapasztalatait. S természetesen a kedves olvasók véleményére, gondolataira is kíváncsiak vagyunk – Ők, Önök hogyan csinálják vagy csinálnák?

A táborok célja és lényeges elvei

Sommásan fogalmazva: ki-ki egyéniségének megfelelően ízelítőt kapjon a számítógépek világáról, aminek alapján eldöntheti, életének mekkora szeletét érdemes erre szánnia. ***Részletesebben:*** minél többen, minél kellemesebben, minél hasznosabban töltsék az időt, és ne kerüljön mindez túl sokba. Természetesen az ellentmondó követelmények még az ésszerűség határain belül is kompromisszumra kényszerítenek.

– Egyszerre egy helyen stábunk kb. 100–150 főt képes fogadni, akik az általános iskola 5–8. osztályát végezték.

– Igyekszünk a táborjelleggel minél inkább ellensúlyozni a tanulást: szakmai programban szinte minden fakultatív. Lehetőséget kívánunk biztosítani, és nem kötelezni.

– A hasznosság tekintetében természetesen „felnőtt" szempontok mérvadóak. Napközben teljesen száműzzük a gyári kalandjátékokat, és a szabadidőben is igyekszünk minimálisra korlátozni. Ilyenek írására csak a legjobbakat bátorítjuk, inkább logikai játékokat ajánlunk. Nagy súlyt fektetünk arra, hogy az elméleti anyagot a meglévő, illetve a kívánatos iskolai ismereteikhez kapcsoljuk.

Az egész tábort a következő elvek vezérlik:

1. Minthogy nem elitképzés a célunk, igyekszünk minél többféle elfoglaltságot ajánlani, mindenki találjon kedvére valót.

2. Minél több csoportot igyekszünk kialakítani. Természetesen az egyéni foglalkozás lenne ideális, de még húszfős társaságok is elfogadhatóak. Időnként kisebb „alcsoportok" képződnek, ami a beosztás dinamikáját szolgálja. A csoportosítás sikerét mutatja, hogy a tábor végére a legjobb csoportban összemosódik az oktató-oktatott szerep.

3. A plusz lehetőségeket (gépidő) ki kell érdemelni, de soha sehol nem kötelező a részvétel!

4. Sűrűn zaklatjuk a résztvevőket „dolgozatokkal". Ez egyrészt állandó visszacsatolás az elmondottakról, másrészt a premizálás és a csoportok közötti vándorlás alapja.

5. Meglehetősen sok a szabadidő, de sokan ilyenkor is lázasan dolgoznak (házi feladat, gondolkodás, szakmai megbeszélés).

6. Sokféle versenyt (programozás, sportvetélkedő stb.) szervez(tet)ünk, motiváló céllal.

Napirend, beosztások, tematika

12 óra általános program (alvás, evés, tisztálkodás, stb.) mellett az egyes csoportok elfoglaltságai:

	VI. (legjobbak)	V.	IV.	III.	II.	I. abszolút kezdők)
gépidő	4	3	2	1,5	1	1
BASIC	0,5	0,5	1	2	2	1,5
angol	1	1	1	1	1	1
logikai j.	1	1	1	0,5	0,5	0,5
előadás	1	1	1	0,5	0,5	0,5
szabadidő	4,5	5,5	6	6,5	7	7,5

– A gépeket napi 7–8 órában osztottuk be, a többi időt (főleg este) jutalmazásra fordítottuk, illetve segítőink kapták. Előfordult, hogy egyszerre több csoport is bent volt, az egészen kezdők közül nem mindig egyesével kerültek komputerhez. Főleg lányok szívesebben alkotnak 2–3 fős „teameket" (esetleg két gépen!). Sok kezdő pedig inkább „nézte" haladóbb társait, tőlük is tanulhattak és szívesebben is kérdeztek.

– A BASIC-ben a gyakorlat (előadás aránya 60–80/20–40 százalék. Rengeteg példát oldottunk meg és ezeket különböző mélységben közösen átbeszéljük. Elsősorban pársoros („utasítás-orientált") programocskák írása. Gyenge kezdőknek (második félidő – I–II. csoport) hasznos gyakorlat, ha egy rövid programot kell elemezni, esetleg hibát keresni! A FOR/STEP elmélyítéséhez igen hasznos segítség a grafika, főleg például C 16-nál különböző látványos ábrák nyerhetők.

– Angolból a BASIC kulcsszavak nagyjából helyes kiejtését tűztük ki célul. Természetesen az angol beosztás független volt a BASIC-től, a nyelvtudás szerint történt!

– Logikai játékok: IV–VI. csoportban szabályismertetés, pici stratégia: GO, malom, amőbák, keresőjátékok (sík, tér), NIM stb. Kezdőknek sok játék elemzéssel: számkitalálósok, keresők, könnyebb táblások, stb.

Előadások:

A számítógép felépítése
Magasszintű nyelvek (LOGO, PROLOG, PASCAL) stb.
Grafikától az animációig
Számítógépek korlátai
A teknőgrafika filozófiája
Beszélgetések választott témákban.

A szerkesztő azért van, hogy a lap olyan legyen, amilyenek az olvasói!

Egy átlagos nap (a harmadik) időbeosztása

	VI.	V.	IV.	III.	II.	I.
reggeli						
8–9						
9–10	Gép-		BASIC	angol		logikai
10–11	idő		gép	esetleg	gép	játék
11–12						
ebéd						
13–14		gép		BASIC		
14–15		gép		BASIC		
15–16 (uzsonna)	E l ő a d á s			gép		
16–17	angol				gép	
17–18		logikai j.				gép
vacsora						
19–22-ig	szervezett szabadidő (vetélkedő, film, disco) jutalom-gépidő, munka, géptől távol } főleg haladók, esetleg bizonyos kezdők kötetlen „hülyéskedés"					

– Az üresen hagyott helyek szabadidőt (sport) jelentenek.
– Szinte TABU a gépek max. kihasználása. Állandó harcot vívtunk a konyhával, hogy lehessen ebédelni és vacsorázni egyenletesen, vagyis sorbanállás helyett dolgozni tudjunk. Legalábbis 10–15 fő állandóan volt a gépteremben 8–21.30-ig. Az oktató középiskolások néha hajnali 2–3-ig dolgoztak, játszottak. Ugyanakkor a géptermi csellengés, hülyéskedés, zavargás könnyen időszakos kitiltással járhat.
– A szabadidő sajnos túl kötetlenre sikerült. A sport: asztalitenisz, labdajáték, fára mászás, fürdés, némi futás.
– Délelőtt megfigyelhető a különböző csoportok géptermi keveredése (IV. cs. 12 fős, csak gyakorolgat; kezdők nézelődnek.)

Induló követelmények és (létszámok) az egyes csoportokban

nap	VI.	V.	IV.	III.	II.	I.
1.	lényegében minden (12)	tömb, FOR, IF, fgv, string (9)	FOR, IF (15)	lényegében semmi (86)		
3.	minden, ami nem gépfüggő, (17)	(19)	FOR, IF tömb, fgv (12)	eddigiek nagyjából (23)	?, =, INPUT (25)	(25)
5.	szinte minden (17)	(19)	nagyjából minden (19)	eddigiek nagyj. (15)	IF FOR nyoma (20)	(28)

– VI. és V. csoport között a harmadik naptól alig van különbség BASIC szempontból. Inkább az érdeklődés, ambíció, gyorsaság szelektál.
– A II. cs. legjobbjai a végére megközelítik a negyedik csoport elsőnapi átlagát!
– A kislétszámú III. cs.-ban a végén nagyobb különbségek lehetnek, mint első nap a 86 kezdő között!
– A sok csoport érvényét általánosítandó: Érdemes akkor is minél több szintet csinálni, ha nincs jelentős különbség!



Hallottam 3–4 érdekes előadást, X. Y-nal még akarok beszélni a ...-ról."
– IV–III. jobbik fele. „Némi fogalmam lett a programozásról, kicsit kevésbé misztikus a számítógép. Kisebb feladatot oldottam meg (pld. számkitalálós, torpedó-adminisztráció, barkochba, stb.)."
– Futottak még. „Kisebb programokat írtam (pld. sorba rendezés, lottóhúzás, egyszerűbb grafika, stb.), néhány logikai játékkal játszottam. Remélem jövőre tovább jutok."
Az utolsó mondat beváltására nagy az esély!
– Leggyengébbek. „Valami csak ragadt rám is..."
Valószínűleg több ragadna az egészen kezdőkre, ha ők is 12–18-an lehetnének. Természetesen bizonyos dolgokat csinálhatnak együtt, de könnyebb a differenciálás, ha minél több fokozat van. A legjobbak nagyobbik fele először vett részt táborainkon, (elő)ismereteit otthon, barátoktól, iskolában szerezte.

Mire van szükség mindehhez?

A megteremtendő szükséges feltételek közül talán a legfontosabb a jó stáb. Háromfős csoportunk 7–8 éves tapasztalatot tudhat maga mögött. Ezenkívül 4–5 rendszeres „profi" segítőnk (kollegák, főiskolások, III–IV. gimnazisták) sok részletben messze felettünk áll. Minden csoporthoz kerül egy-egy ilyen oktató, illetve még 1–2 kevésbé felkészült középiskolás, aki tavaly esetleg csak IV. csoportos volt! Utóbbiak elsősorban tanítani tanulnak, persze saját ügyükben is kérdezhetnek – ezt főleg egymástól teszik. Nyugodtan tekinthetnénk őket VII. csoportnak, kevés szabadidővel! Nem árt, ha az egyéb elfoglaltságoknak is van külön felelőse.

BASIC tematika az egyes csoportokban

nap	VI.	V.	IV.	III.	II.	I.
1–2	Csak gyakorlat, hibák megbeszélése, gépkönyvek		ismétlés	alapfogalmak PRINT, LET, INPUT,		pici FOR, IF
3–4	gondolkodás, önálló feladatokon, gépidő alatt szükség szerint segítünk.		gyakorlás	IF, FOR, RND	ismétlés,	IF
5–8			aki akar, egyszerűbb feladatokat old meg önállóan	függvények, $, pici grafika, tömbök	ismétlés, picit tovább	ismétlés program használat

– Célunk annyi BASIC-et tanítani, hogy utána a gyerekek egy (rosszul megírt) gépkönyvben is eligazodjanak. A többi már „magától" megy!

Ezek után amit egy átlagos csoporttag elmondhat a tábor végén:

– V–VI. csoport. „Írtam önállóan egy nagyobb programot (például hívólift szimuláció, síelő, malom, stb.). A 70–150 soros programom 75–99%-ban el is készült, jól működik.

– A szükséges gépparkot (30–35 komputer, azaz 3–4 fő gépenként) nagyrészt kollégáktól, ismerősöktől kapjuk kölcsön, 8–10 darabot iskoláktól, és néha a gyerekek is hoznak magukkal. Napi 12 órás üzemidőt számolva ez sok is.

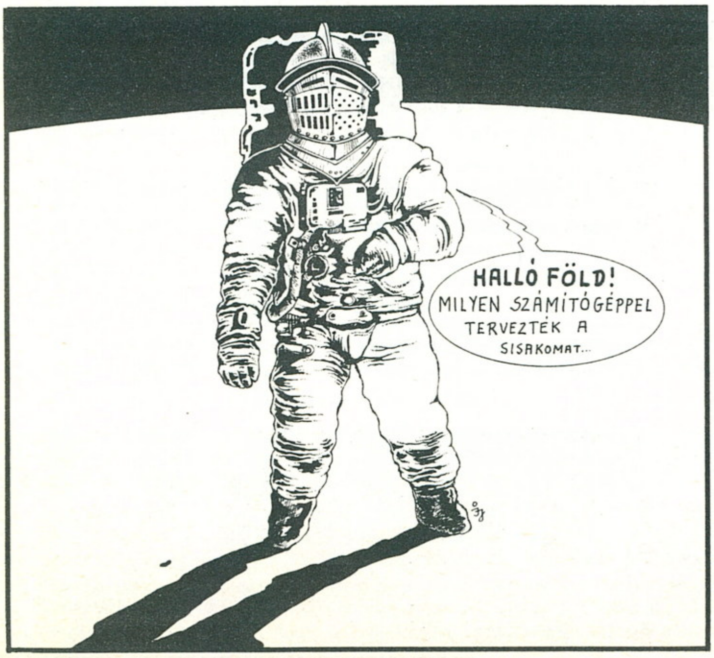
A gépparkból számszerűleg messze kiemelkedik a Commodore (PET, VC 20, 16, 64), sok még a HT. A gyerekek imádják a Spectrumot – mi kevésbé: sok baj van vele. A Sinclair gépeken kívül az utolsó négy évben egy-egy VC 20 és C 64 típus romlott el nálunk. Igyekszünk minden géphez saját tartozékait ren-

delni (tv, kábelek, perifériák) és ezeket jól megjelöljük. Gépet (és tartozékot) átdugaszolni, kikapcsolni csak oktatóknak szabad, és erre figyelünk is.
– Érdemes a BASIC mellett mást is tanítani. A logikai játékokat azért erőltetjük, mert igen hasznosak az algoritmusfogalom kialakításában, megértésében. Elképzelhető még valami manuális képzés is (rajz, papírhajtogatás stb.) – főleg alacsonyabb csoportokban.
– Természetesen elsősorban saját kiadványainkat szeretjük. (Tankönyvek, példatárak, ötlettár, stb.) Ezeken kívül örömmel használjuk a Hetedhét sorozat (NOVOTRADE) köteteit és a sz. gépes játékokról szóló könyveket (Csákány–Vajda, Spencer, Illetve Z. Nowak: 50 táblás játék).
Ajánlani merjük továbbá:
D. Alcock: Ismerd meg...; különösen a nyelvjárásokkal!
Appal–Kőhegyi–Zsakó: Számítógépes feladatok (példatár)
INTERPRESS; Dusza–Varga: A BASIC nyelvű programozás
Lőcs Gy.: A BASIC és a Kíváncsi (reméljük nyárra a példatár is megjelenik!)
Előadásokhoz jó ötletek meríthetők a Műsoron a számítógép c. könyvből, illetve a µ Magazin és BIT-LET számaiból. Ilyen előadásokra nagyobb (haladóbb) gyerekek is vállalkozhatnak – előzetes felkészülés után.

További tanulságok

– Fel kell készülni arra, hogy a résztvevők 10–15%-ára igen kevés hatással van az egész. Az utolsó 2–3 évben ennek megváltoztatása volt az egyik fő célunk (elsősorban ezért ragaszkodunk a komplex táborhoz), sajnos igen kis sikerrel. Az immunisak nagy részéről elmondható, hogy szinte semmi sem érdekli őket, se BASIC, se sport, talán még a film és a disco sem. Általában úgy kerülnek ide, hogy szüleik néhány nap szabadságot akarnak s éppen ez a tábor esik látókörükbe. Létüket tudomásul kell venni, továbbra is igyekszünk csökkenteni a számukat.
– A résztvevők többsége szerint minden gépidő kevés! Azzal szoktuk őket reálisabb véleményre bírni, hogy érdeklődünk a gép előtt, illetve a gondolkodással, tervezéssel töltött idő arányáról. Utóbbi időt elenyészőnek vallják maguk is. Főleg kezdőktől, de eleinte mindenkitől megköveteljük, hogy a gép elé olyan papírral üljön le, amin már nyoma van a beírandók átgondolásának.
– Nagyon kevés segítséget kapunk – még haladóktól is –, hogy nekik való, érdekes feladatokat találjunk ki. Túl könnyen, kritikátlanul egyeznek bele a javaslatainkba, és azután nagy a kiábrándulás.
– Érdekes, hogy alig vehető észre egy kezdőről, hogy milyen mértékben férhetett hozzá számítógépéhez a tábor előtt (otthon, rokonnál, ismerősnél, stb.). Másként fogalmazva meglepő, hogy ez a korosztály magától mennyire nem képes elindulni az igazi ismerkedés útján. Persze kivételek akadnak. Sokkal jellemzőbb azonban, hogy egy I–II. csoportbeli gyerekre következő évben rá sem ismerünk, azonnal „szuper" lesz! Ezt feltétlenül saját sikernek is érezzük, persze döntő az évközbeni gyakorlás is.

Török Turul



NYÍLT TÉR

Nagy öröm volt olvasni Angyalosi László sorait a BIT-LET 1986. január 28-i számában.

Dühöngéseit maximálisan megértem és egyet is értek vele, hisz a C 16 7900 forintos ára a magyar piacon szenzációként hatott, de sokat nem ér, ha a folytatás elmarad és csak ízelítőnek szánták.

Feltétlenül – a tisztánlátás érdekében – közölni kell a nagyközönséggel, hogy a Tudományszervezési és Informatikai Intézet elmarasztalása, hogy drágán vásárolt vagy „egy kereskedelmi cég közvetítésével" ugyanaz a termék 20%-kal többe kerül – nem helytálló.

Először is a C 16-osok szállítására vonatkozó szerződés megkötésére az iskolák részére lényegesen korábbi időpontban került sor, amikor is a listaárak magasabbak voltak, mint az ÁPISZ vásárlásának időpontjában, tehát eltérő devizaárakról van szó.

Az ár három részből tevődött össze, a Skála-Coop szállította az alapkonfigurációt – viszonylag olcsón – a Novotrade pedig vállalta a garanciát, az alkatrészellátását, biztosított egy „Bevezetés a Basic"-be c. könyvet magyar nyelven, 2 db hozzá tartozó, magyar nyelvre adaptált szoftverrel, és 1 db demokazettát magyar nyelvű leírással, 3 db magyar nyelvű felhasználói kézikönyvet, 1 db C 16 Hetedhét könyvet, egy Basic-emlékeztetőt, amelyet a klaviatúrára lehet ráhelyezni, továbbá a C 16 gép a magyar szabványnak megfelelő hazai körülmények között kialakított klaviatúrával és a megfelelő – speciálisan a magyar piacra készített – a magyar ABC-t tartalmazó ROM-mal került az iskolák birtokába, úgy hogy minden egyes berendezés átadás előtt bevizsgálásra került.

Ezenkívül a TII hosszasan és sokoldalúan vizsgálta a berendezés hazai elterjesztésének összes feltételeit, így elvégeztette a MEEI és a Posta kötelező vizsgálatait, valamint alaposan ellenőrizte az összes dokumentációs anyagot. Összegyűjtötte az iskolák megrendeléseit és saját eszközeiből ingyen adott gépeket az iskoláknak a fizetők mellett.

Mindezt természetesen az ÁPISZ-nak nem kellett megtennie, viszont az általa forgalomba hozott gép *nem* magyar klaviatúrás, *nem* tartalmazta mindazokat a szolgáltatásokat, amelyeket az iskolák kaptak.

Mindezeket összevetve a tényleges árkülönbség minimális volt és én nem azt feszegetném eltérő műszaki szolgáltatásoknál, hogy 7900 Ft vagy 8775 Ft, hanem azt, hogy miért nincs egyáltalán egy darab sem. Az üzletek tele vannak hazai, szocialista és tőkés hi-fi-berendezésekkel, szórakoztató elektronikával és nyugati márkás élvezeti cikkekkel, de ami a jövőnket alapvetően érinti az nincs; a számítógép mintha *tabu* lenne az ÁPISZ-on és a Skála-Coop-on kívül a belkereskedelemnek. Ezen kellene változtatni és elgondolkozni.

Mellesleg itt jegyzem meg, hogy a Novotrade 3 hónappal ezelőtt megrendelt az iskolák részére további 3500 db számítógépet a Metrimpexnél és a szükséges importengedély illetékét is befizette.

Ide kívánkozik az az információ is, hogy a 2C üzlethálózatban (ÁKV, Művelt Nép, Novotrade és az ÁPISZ-SZÁMALK szaküzletekben) ezrével adtak el először Magyarországon *olcsó* szoftvert, melynek a fogyasztói ára 220 Ft volt.

Reméljük, heteken belül egy sor új szoftvertermékkel és főleg olcsó termékkel találkozhatnak a C 16 tulajdonosok.

Üdvözlöm a szerkesztőt és további jó egészséget kívánok *mindazoknak*, akik részt vettek a C 16 bevezetésében Magyarországon.

Rényi Gábor, ügyvezető igazgató, Novotrade Rt

Köszönjük Rényi Gábor hozzászólását. Mi őszintén szólva nem a Novotrade-nek, s nem is a TII-nek címeztük január végi írásunkat. De hát akinek inge – úgy tűnik, rossz helyre címeztünk. Véleményünk nem mindenben változott. Az árakban lehetséges, hogy tévedtünk – így hát nem 20%-kal, csak bő 10-zel fizettek többet az iskolák azokért a gépekért, amelyekre a fizetést követően még hónapokat vártak, várakozásuk közben megérkeztek és elfogytak a gépek az ÁPISZ-ban. Mindezért megütötte őket a guta. Magyar nyelvű felhasználói kézikönyvet, demokazettákat az ÁPISZ is adott a gépekhez, a magyar szabvány szerinti billentyűzetről pedig csak annyit: nem tudjuk, milyen magyar szabványra gondol Rényi Gábor, de rég láttunk olyan rosszul elrendezett magyar billentyűzetet, mint ez. A közlés további részével egyetértünk, s köszönjük az információkat.

```
10 PRINTCHR¤(2): CLS:PRINT¤5,0,"   --- MON
ITOR ---"
11 PRINTCHR¤(1):PRINT¤7,9,"NAGY'BETÜKET'HA
SZNÁLJON!!":FORI=0TO1000:NEXTI
12 PRINT¤10,10,"KéR TÁJéKOZTATOT? (I/N)"
13 IF INKEY¤=""THEN13
14 IFINKEY¤<>"I"THEN100
15 CLS
16 PRINT¤5,2,"EZ A SEGéDPROGRAM ALKALMAS A
RRA,HOGY     ÁTTEKINTSEN EGY-EGY MEMORIATER
ÜLETET,      FOLYAMATOSAN VAGY TETSZŐLEGES
LéPéS-       NAGYSÁGGAL LéPTESSEN ELŐRE IS
,HÁTRA       IS.
20 PRINT¤10,2,"A MEMORIACIMEK TARTALMÁT MO
DOSITHATJA,   AKÁR DECIMÁLIS,AKÁR HEXADECI
MÁLIS BEIRÁS-  SAL"
30 PRINT¤14,12,"TOVÁBB:<BILLENTYÜ>"
35 IFINKEY¤=""THEN35
40 CLS:PRINT¤2,12,"H A S Z N Á L A T A :"
50 PRINT¤4,3,"CiM?-RE ADATBEVITEL:"
55 PRINT¤5,6,"0-32767            decimális
              OH-7FFFH     hexadecimális
                             értékben és ala
kban"
60 PRINT¤9,3,"LéPTETéS:                +     e
lőre                                   -     h
átra"
70 PRINT¤11,3,"CIM TART.MODOSITÁS:- I - eg
yszeri                             - M - fo
lyamatos"
80 PRINT"                          - x - vissz
atérés"
90 PRINT¤14,12,"TOVÁBB:<BILLENTYÜ>"
95 IFINKEY¤=""THEN95
96 CLS:PRINT¤5,5,"M-RE VAGY I-RE ADATBEVIT
EL:                 0-255 VAGY OH-FFH FORMÁB
AN."
97 PRINT¤14,12,"TOVÁBB:<BILLENTYÜ>"
98 IF INKEY¤=""THEN98
100 CLS:INPUT"CiM";D¤:L=0:J=0:U=0:A=0:D=0:
Q=0
101 H=LEN(D¤):FORI=1TOH-1:B¤=MID¤(D¤,I,1):
B=ASC(B¤):IFB<48ORB<65ANDB>57ORB>70THEN100

102 NEXTI:IFH>5THEN100
103 IFASC(MID¤(D¤,H,1))>57ANDASC(MID¤(D¤,H
,1))<>72THEN100
104 INPUT"LéPéSNAGYSÁG";W
105 REM
110 L¤=RIGHT¤(D¤,1):IFL¤="H"THENGOSUB1000
120 IFL¤="H"THEN146
130 D=VAL(D¤)
135 IFD>32767THEND=D-65535
145 IFL¤="H"THENGOSUB5000
146 IFL¤="H"THENA=D+1ELSEA=D
148 IF A<0ORA>65535THENA=0
150 U=PEEK(A)
155 REM      ******* CIM TART. HEXÁBA *****
*

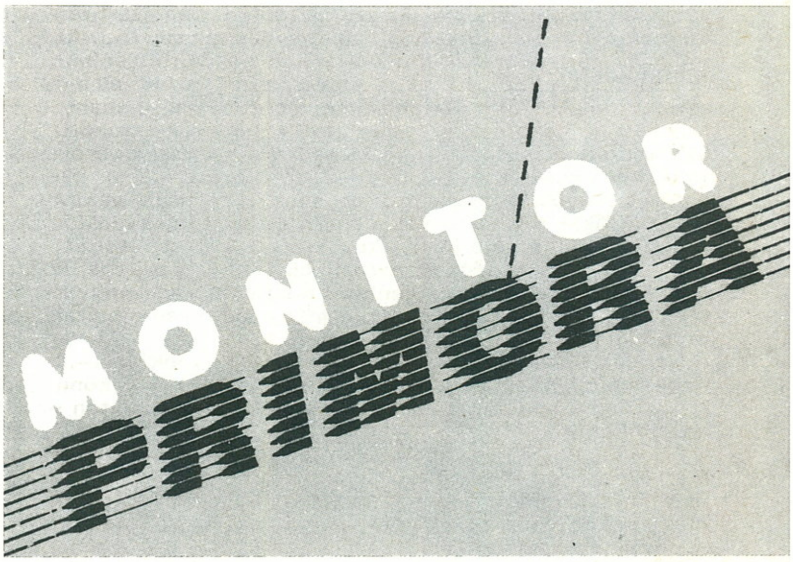
160 V=U:H¤="":FORB=1TO0STEP-1
170 N=INT(V/16üB)
180 H¤=H¤+CHR¤(N+48-(N>9)*7)
190 V=V-N*16üB
200 NEXTB
490 REM
496 REM          *** KIÍRATÁS ***
498 REM
500 IF L¤="H"THENGOTO515
502 LE=0
503 GOSUB5000
504 X¤=D¤:LE=1
505 PRINT"H  =  "D;"  =  "H¤;"H";:GOTO520
515 I¤="":PRINT"  ";LEFT¤(D¤,4);"H  =  ";D
;"  =  ";H¤;"H";
520 PRINTTAB(25)"  =  ";RIGHT¤(""+STR¤(U),
3);
530 IFU>30ANDU<152THENI¤=CHR¤(U)ELSEI¤=""
540 PRINTTAB(35)"  =  "I¤
541 IFS¤="I"ORS¤="M"THENRETURN
543 REM
544 REM     ***** CIM LéPTETéSE; MODOSiT **
*
545 REM
550 S¤=INKEY¤:IFS¤="+"THEN2000
560 IFS¤="-"THEN2100
570 IFS¤="I"THENGOSUB2200
575 IFS¤="X"THEN100
578 IFS¤="M"THENGOSUB10000
580 GOTO550
990 REM
993 REM  ***** HEXA CIM DECIMÁLISBA! *****

995 REM
1000 REM
1010 L=1:R¤=D¤:N(I)=0:P=0:X=LEN(R¤)
1020 FORI=1TOX-1:E¤(I)=MID¤(R¤,I,1)
1030 IFASC(E¤(I))>57THEN1500
```

A BIT-LET PRIMO vallatójában leírak szerint a gépi kódú programozhatóság az átlagos felhasználó számára még nehézkes.
Ennek egyik alapvető oka, hogy nincs a gépbe épített monitorüzemmód, amellyel vizsgálható, áttekinthető az egész memóriaterület és a RAM-ban adott esetben folyamatos módosítás érhető el.
Más gépnél (VC 20) már találkoztunk olyan BASIC-ben írt monitorprogrammal, amely az előbbi igényeknek eleget tesz.
A szentesi Deák Ferenc utcai Általános Iskola jóvoltából általános iskolás fiammal együtt hozzáférhetünk az iskola A 32-es típusú gépéhez. Ez a lehetőség és az előbb említettek adták az ötletet, hogy megírjunk egy egyszerű, de – rövid tapasztalatunk alapján is állítjuk – jól használható BASIC monitorprogramot a PRIMORA.
A hexadecimálisról decimálisra és vissza számítások matematikáját és a program elvi felépítésének lényegi lépéseit én, a konkrét megírást és megjelenítést a fiam oldotta meg.

A program a következőket tudja:

A tájékoztató utáni **CIM?**-re a vizsgálandó memóriaterület kezdőcímét kell beírni. Akár decimális, akár hexadecimális



beírás lehet, csupán ez utóbbi esetben az pl. 7500H alakú legyen. Ha a **léptetés?**-re 1-et írunk, akkor a léptetés folyamatos lesz, de bármilyen érték használható, ha sietősebb és nagy vonalakban történő megfigyelést végzünk.
5700H cím esetén a megjelenítés pl.
5700H = 22272 : 41H = 65 = A
Az utolsó helyen a cím tartalmának CHRS értékét írja ki, ha az megjeleníthető.
Előre léptetés a +, hátra léptetés a – billentyűkkel történik.
Az I lenyomása után az adott címre 00H–FFH vagy 0–255 formátumban egyaránt írhatunk. Ha folyamatos változtatást akarunk, az M billentyű lenyomásával indíthatjuk azt. Kilépés az X billentyűvel történik. Használjunk nagybetűket!
Programunk, bár nem túl gyors, de úgy hisszük könnyen kezelhető.
Segítségével és a BIT-LET-ből nyert apróbb-nagyobb információkkal elég sok konkrét adatot „felfedeztünk".
Ezek lényegesebbjei:

1. A BASIC program kezdete: 43E9H (17385)
vége: 67FFH (26623)

2. Képernyők memóriakezdete: 6800H (26624)
vége: 7FFFH (32767)
A 40B1 és 40B2 címeken a képernyő memória kezdőcíme van elhelyezve. A kezdőcímtől folyamatosan bevitt megjelenítési adatokat balról jobbra egymás mellé írni. Egy sorban 32-t és 192 sort összesen.

3. A 4044H (16452) címre tárolódik az éppen lenyomot billentyű belső kódja. Ennek előírása pl.

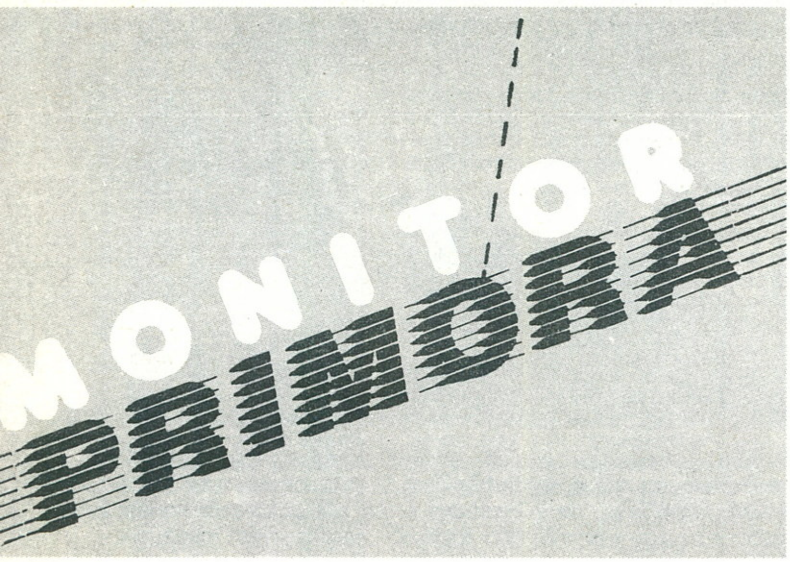
```
10 CLS: AS = INKEYS          Pl. A  142
20 IF AS = ""THEN 20             B  154
30 ?AS; PEEK (16452)             C  144
40 GOTO 20                          stb.
```

Ha mozgatásra kívánjuk használni, AS = INKEYS deklarálása után működik.

4. 4423H (17443) címen tárolódik az INKEYS függvény aktuális értéke. A billentyű nincs lenyomva: 0
Lenyomott billentyű: 1

5. A BASIC kulcsszavaknak billentyűkről történő beviteléhez szükséges adatokat egyszerűen elő lehet hívni. Pl.
10 REM A REM után a billentyűket szépen sorban egyedül, a ↓-lal együtt, majd a CTR és ↓-lal együtt nyomjuk le. Érdekes grafikus jelek adódnak, amelyek helyett listázás után a billentyűhöz tartozó kulcsszavakat láthatjuk. Természetesen ez a lehetőség editálás közben elvész.

6. Tiszai Tamás cikke igen jelentős segítség az új karakterek generálásához. Kár, hogy a konkrét cím adatok közlésénél meglehetősen visszafogott.
Megnéztük, hogy az általa említett 16459–460-as címek alaphelyzetben milyen címre mutatnak. Ez 3507H (13575 nek adódott.
E cím környékét a monitorprogrammal átvizsgálva megtaláltuk a gép karaktereinek pontmintázatát tartalmazó címterületet.
Ez 31F4H (12788) – 35BF (13759) terjed.



Pl. a 31F4 címet követő 12 byte adja az í betűt. Örömmel fedeztük fel a 3208H-t követő 7 byte CHRS megjelenítésénél a TISZAI nevet. (Bizonyára Tiszai Tamás alkotta a gép BASIC-jének ezt a részét!)
Pl. a 8-as szám pontmintázatának a 13000–13007 címeken található decimális adatai 56,68,68,56,68,57,0.

7. A HT 1080 Z gép meglévő memóriacímeit mankónak használva vettük észre azt is, hogy a 40A7–40A8 címek a PRIMONÁL is a 41E8H-ra mutatnak, amelytől a HTZ-n az egyes regiszterek pillanatfotóit tartalmazó byte-ok következnek.
Hasonló lenne a PRIMÓNÁL is a szervezés?
Ezeken kívül még sok, de egyelőre csupán kecsegtető adatunk van.

REM a HT gépnyerő pályázati anyagokról írt kritkát olvasva, illendően megszeppenve küldjük el „rendetlenül" sorszámozott programunkat, de illedelmes szépítési szándékunkat keresztülhúzta az általunk igen nagyrabecsült PRIMO gép azzal, hogy néhány billentyűje „elefánt" üzemmódra váltott és szervizbe kellett adni.
Gép nélkül nem mertük komolyabban megbolygatni a jól működő programot.

ifj. Gulyás László tanuló, Gulyás László tanár

```
1040 N(I)=VAL(E¤(I))
1044 M=N(I)*16ü(4-I):P=M+P
1047 NEXTI
1060 D=P
1068 U=PEEK(D+1)
1070 RETURN
1500 C=ASC(E¤(I)):N(I)=C-55
1510 GOTO1047
2000 D=D+W:GOTO145:REM  ** ELŐRE LéP! **
2100 D=D-W:GOTO145:REM  ** HATRA LéP! **
2190 REM
2192 REM    ***  ADATBEVITEL **
2194 REM
2200 X¤=D¤:PRINT;LEFT¤(X¤,4);"H";: PRINTS¤
;:INPUTU¤:IFLEN(U¤)>3THENPRINT"??":GOTO220
0
2202 H1=LEN(U¤):FORI=1TOH1:B1¤=MID¤(U¤,I,1
):B1=ASC(B1¤):IFB1<48ORB1<65ANDB1>57ORB1>7
0ANDB1<>72THEN2204
2203 NEXTI
2204 IFH1=3ANDB1<48ORB1<65ANDB1>57ORB1>70A
NDB1<>72THENPRINT"??":GOTO2200
2205 Z¤=RIGHT¤(U¤,1)
2210 IFRIGHT¤(U¤,1)="H"THEN6000
2220 U=VAL(U¤):IFU<0ORU>255THENPRINT"??":G
OTO2200
2225 GOSUB155
2226 Z¤=RIGHT¤(U¤,1)
2230 IFL¤="H"THENPOKED+1,UELSEPOKED,U
2240 RETURN
4990 REM
4994 REM        ****  ATSZAMITAS HEXABOL ***

4996 REM        ****      DECIMALISRA!    ***

4998 REM
5000 G¤=""
5003 V=D
5005 FORB=3TO0STEP-1
5010 M=INT(V/16üB)
5028 G¤=G¤+CHR¤(M+48-(M>9)*7)
5030 V=V-M*16üB:NEXTB
5040 D¤=G¤
5042 IFL¤<>"H"ANDU¤<>"X"ANDZ¤="H"ANDS¤="M"
ANDLE=0THENPRINT;X¤;:GOTO5047
5043 IFL¤<>"H"ANDZ¤="H"ANDS¤="M"THENGOTO50
47
5045 IFL¤="H"THENPRINT"";ELSE PRINT;D¤;
5047 Z¤=""
5050 RETURN
5053 RETURN
5990 REM
5992 REM      ***** CiM TART.HEXA-DEC. ATV.
***
5994 REM
6000 D¤=LEFT¤(U¤,2):X=LEN(D¤):P=0
6010 FORI=1TOX:T¤(I)=LEFT¤(RIGHT¤(D¤,I),1)

6020 IFASC(T¤(I))>57THEN6500
6030 N(I)=VAL(T¤(I))
6040 M=N(I)*16ü(I-1):P=M+P
6050 NEXTI
6055 U=P
6056 D¤=X¤
6058 IF Y=1THENGOTO10050
6060 GOTO2225
6500 C=ASC(T¤(I)):N(I)=C-55
6510 GOTO6040
7000 T=1:RETURN
7990 REM
7992 REM      **** LéPTETéSES ADATBEVITEL **
*
7994 REM
10000 X¤=D¤:PRINTS¤;:INPUTU¤:IFLEN(U¤)>3TH
ENPRINT"??":GOTO10000
10002 IFL¤="H"THENQ=1ELSEQ=0
10004 IFU¤="X"THEN100
10005 H2=LEN(U¤):FORI=1TOH2:B2¤=MID¤(U¤,I,
1):B2=ASC(B2¤):IFB2<48ORB2<65ANDB2>57ORB2>
70ANDB2<>72THEN10008
10007 NEXTI
10008 IFH2=3ANDB2<48ORB2>57ANDB2<65ORB2>70
ANDB2<>72THENPRINT"??":GOTO10000
10010 IFU¤="X"THEN100
10020 Z¤=RIGHT¤(U¤,1)
10030 IFRIGHT¤(U¤,1)="H"THENY=1:GOTO6000
10040 U=VAL(U¤):IFU<0ORU>255 THENPRINT"??"
:GOTO10000
10050 Y=0:LI=0:GOSUB 155
10060 Z¤=RIGHT¤(U¤,1):LI=0
10070 IF L¤="H"THENPOKED+1,UELSEPOKED,U
10080 D=D+W
10090 GOSUB5000
10100 GOTO10000
10200 GOSUB155
10210 RETURN
```

„TEDMON" MONITOR



A TEDMON a C16-ba beépített gépi kódú monitorprogram. Ez tartalmaz egy miniassemblert és egy disassemblert is. Mindezek kezelése az angol és a magyar nyelvű gépkönyvből hiányzik – bár a megfelelő fejezetre van utalás –, így ezt most közzétesszük.

Utasítások összefoglalása

A *Assemble:* Lefordít egy mnemonikban írt programot a 7501 processzor gépi kódjára
C *Compare:* Összehasonlít két memóriaterületet és jelzi az eltéréseket
D *Disassemble:* Visszafordít egy gépi kódú programrészt mnemonikokra
F *Fill:* Feltölt egy memóriaterületet egy adott értékkel
G *Go:* Elindít egy gépi kódú programot az adott címtől
H *Hunt:* Átfésül egy memóriaterületet, és kijelzi azokat a címeket, amelynek tartalma megegyezik egy előre adott értékkel
L *Load:* Programot tölt be kazettáról vagy diszkről
M *Memory:* Egy memóriaterület tartalmát írja ki
R *Registers:* Kijelzi a processzor regisztereinek aktuális tartalmát
S *Save:* Elment egy programot kazettára vagy diszkre
T *Transfer:* Blokkmásolás
V *Verify:* Összehasonlít egy, a memóriában levő programot a kazettán vagy diszken levő programmal
X *eXit:* Visszatérés BASIC-be
. *(pont):* Egyenértékű az „A"-val
〉 *(nagyobb):* Adott címtől legfeljebb 8 byte értékét változtathatjuk
; *(pontosvessző):* "**R**" utasítás után regiszterek értékét lehet változtatni

A monitor által kiírt szöveget általában közvetlenül fel lehet használni egy másik utasítás beírásához (pl.: memóriatartalom megváltoztatása "**M**" utasítással).

TEDMON kezelése

Belépés: **MONITOR** paranccsal
Belépéskor kiírja a regiszterek aktuális tartalmát (éppúgy, mint az "**R**" utasításnál), majd egy villogó kurzor jelzi, hogy a monitor kész parancsok fogadására.
Az utasításokban a paramétereket – ha nincs külön jelölve – betűközökkel kell elválasztani.

Utasítások részletes ismertetése

A Szintaxis: **A** *cím mnemonik operandus*
cím: hexadecimális szám, amely címre a megfelelő gépi utasítás kerül.
mnemonik: pl. LDA
operandus: a megadott mnemoniknak megfelelően (néhánynál operandusra nincs szükség)
A befejezett utasítást *RETURN*-nal kell lezárni. Hiba esetén egy kérdőjel kerül a sorba. A kurzor a következő sorra lép. Helyes bevitel esetén a következő sor elején a monitor kiír egy A betűt és a következő címet. Pl.:
A 2000 LDA $00 *(RETURN)*
A 2002
Az "**A**"' utasítással egyenértékű a pont.
C Szintaxis: **C** *cím1 cím2 cím3*
cím1: hexadecimális szám, az összehasonlítandó blokk kezdőcíme
cím2: hexadecimális szám, az összehasonlítandó blokk végcíme
cím3: hexadecimális szám, a másik összehasonlítandó blokk kezdőcíme
Különbségek esetén kiírja azokat a címeket, amelyek tartalmai különböznek
D Szintaxis: **D** *cím1 cím2*
cím1: hexadecimális szám, kezdőcím
cím2: hexadecimális szám, végcím
Ha cím2 elmarad, akkor egy teljes képernyőnyi részt fordít vissza. Ha cím1 is elmarad, akkor a kezdőcím az előző utasítás végrehajtásakor utoljára kiírt címnél 1-gyel nagyobb. Csak "**D**"-t begépelve lapozni lehet. Pl.:
D 3000 3004 *(RETURN)*
. 3000 A9 00 LDA # $00
. 3002 FF ???
. 3003 D0 2B BNE $3030
D *(RETURN)*
. 3005 . . .
. . . .
F Szintaxis: **F** *cím1 cím2 byte*
cím1: hexadecimális szám, a feltöltendő blokk kezdőcíme
cím2: hexadecimális szám, a feltöltendő blokk végcíme
byte: egy- vagy kétjegyű hexadecimális szám, amely értékkel fel kívánjuk tölteni a kijelölt blokkot. Pl: F 0400 0500 FE
G Szintaxis: **G** *cím*
cím: hexadecimális szám, amelytől a programot futtatni akarjuk. Ha a cím elmarad, akkor az "**R**" utasítás által kiírt PC értéktől indul a program. A program végére ezen utasítás használata esetén nem a szokásos RTS utasítást kell elhelyezni, hanem BRK-t.
H Szintaxis: **H** *cím1 cím2 adatok*
cím1: hexadecimális szám, a blokk kezdőcíme
cím2: hexadecimális szám, a blokk végcíme
adatok: hexadecimális számok vagy ASCII karakterek. (A kettő nem keverhető.) Az egyes adatok közé betűközt kell írni. Az ASCII karakterek elé aposztrófot. Pl.: H C000 FFFF 'READ
L Szintaxis: **L** *file-név, eszköz*
file-név: ASCII karaktersorozat idézőjelek között
eszköz: 1=magnó, 8=floppy
Kezdőcímet nem lehet megadni, mert a program (adatblokk) mindig oda kerül vissza a memóriába, ahonnan azt felvettük. Pl.:
L "PROBA",1
M Szintaxis: **M** *cím1 cím2*
A két cím megadása azonos a "**D**" utasításnál leírtakkal. A memóriatartalom a következő formában jelenik meg: Pl.
M A048 A050 *(RETURN)*
〉A048 41 E7 00 AA AA 00 98 56 :<u>*A!.**..V*</u>
Az aláhúzott karakterek inverzben jelennek meg. A memóriatartalmat egyszerű felülírással lehet változtatni.
〉 Szintaxis: **〉** *cím adatok*
cím: hexadecimális szám, erre a címre íródik az első adatbyte
adatok: legfeljebb 8 db kétjegyű hexadecimális szám, köztük betűköz. Pl.:
〉3000 23 45 56 *(RETURN)*
A $3000 címre íródik a $23, a $3001 címre a $45, stb.
R Szintaxis: **R**
Pl.:
R *(RETURN)*
PC SR AC XR YR SP
;3020 00 20 00 E0 78
A pontosvessző után megadott hexadecimális számokkal a regiszterek tartalmát változtatni lehet.
S Szintaxis: **S** *file-név, eszköz, cím1, cím2*
file-név: ASCII karaktersorozat idézőjelek között
eszköz: 1=magnó, 8=floppy
cím1: kiviendő blokk kezdőcíme
cím2: kiviendő blokk végcíme
Pl.: S "JATEK",8,0400,0BFF
T Szintaxis: **T** *cím1 cím2 cím3*
cím1: átmásolandó blokk kezdőcíme
cím2: átmásolandó blokk végcíme
cím3: célcím, ide kerül az előzőleg kijelölt blokk
Ha a két címtartomány részben átfedi egymást, akkor adatveszteség léphet fel, így célszerű ebben az esetben a blokkot először egy „semleges" területre másolni.
Pl.: T 1400 1600 3400
V Szintaxis: **V** *file-név, eszköz*
A paraméterek magyarázatát ld. "**L**" utasításnál.
Pl.: V "PROBA",8
X Szintaxis: **X**
Visszatérés BASIC-be

Stranz Jan-Marc, Halász Péter

ötlet

BASIC LOGO



Visszaemlékezve a C 16 vallatására, a a gép programnyelve impozáns egyhangúsággal kapott 5,0 osztályzatot. Hasonló sikerrel csak a C 64 billentyűzete dicsekedhet, és megközelítő átlagok sem hemzsegnek. Sőt, jóllehet Simonyi Endre kimerítően elmarasztalja ezt a géptípust (μMagazin 85/6), a BASIC-jéről ő sem tud rosszat mondani: ebből a szempontból – úgy tűnik – még a DRAGON-nal is felveszi a versenyt, ami gondolom ekvivalens a számítástechnikai Nobel-díjjal.

Nos, miért vajon mindez? Számos tényező mellett az alábbiak szempontjából a legfontosabb, hogy tudja szinte a teljes LOGO (teknőc)-grafikát. Potenciális iskolaszámítógépről lévén szó, ez rendkívül fontos és ezért sietek megosztani tapasztalataimat, melyekről a gépkönyvek diszkréten hallgatnak.

A teknőc világhódító karrierje abban keresendő, hogy segítségével egészen kis gyerekek is nagyon látványos ábrákat készíthetnek. A hagyományos, derékszögű (Descartes-féle) koordinátarendszerekben gondolkodó BASIC-ekkel szemben ez a gép polárkoordinátákkal is tájékozódik a síkon. Így nagyon sok dolgot frappánsan egyszerűen old meg, amit a többiek csak „gusztustalan trigonometrikus bűvészkedéssel" képesek. Zátonyi Sándor kollégám már említette ezt a lehetőséget (BIT-LET 86' január), az ő felfedezését szeretném kicsit továbbgondolni.

Mik a polárkoordináták?

A síkon például úgy is egyértelműen tájékozódhatunk, ha egy rögzített origóból kiindulva megmondjuk, hogy pontosan milyen irányba (egy szög), és mennyit (egy hosszúság) menjünk. Ez a teknőc filozófia alapja: (menj) ELŐRE és FORDULJ! Ez teszi lehetővé, hogy a LOGO-ban trigonometrikus függvények, sőt négyzetgyökvonás nélkül kört rajzoljunk, vagy egy nem feltétlenül szabályos alakzatot elforgassunk.

A szóban forgó utasítás pontos szintakszisa:

DRAW 1 TO R;S vagy DRAW TO R;S

Hatására a PC (pixel cursor) pillanatnyi helyzetétől S foknyira és R távolságra lévő pontba húz egy vonalat a gép a PC-ből. Az S=0 alapértelmezése a függőlegesen felfelé mutató irány, S=90 esetén tehát (vízszintesen) jobbra mehetünk. Egyetlen szépséghiba, hogy S nem relatív adat, hanem minden esetben az S=0 etalonhoz méretik.

1. PROGRAM

```
100 GRAPHIC 1,1
120 FOR I=0 TO 360 STEP 30
130 DRAW 1,160,100 TO 95;I
150 NEXT I
```

EZ IS LEHET!
EKVIVALENSÁ
125 LOCATE 160,100
130 DRAW TO 95; I
MEGOLDÁSSAL

SZÁMÍTÁS ALAPJÁN
ELŐSZÖR 60,100
VOLT, DE KORRIGÁLNI
KELLETT!

2. PROGRAM

```
200 LOCATE 67,103
210 FOR I=0 TO 360 STEP 5
220 DRAW 1 TO 8;I
230 NEXT I
```

Fenti programocskák eredményei sokkal egyszerűbben is előidézhetők C 16-on, pillanatnyi célunk azonban a pontosvessző használatának bemutatása volt.

BASIC a LOGO nyomában ... Sőt!?

Jogosan méltatja több cikk a teknőcgrafikát a hazai „szaksajtóban" is (elsősorban Ury L.: μ Magazin 85/1., Szekfü A.: BIT-LET 1983. október és 1984. április, Zátonyi S.: BIT-LET 1984. (augusztus és Szabó Gál A.: BIT-LET 85. július).

Az alábbiakban néhány már ismert LOGO-látványt reprodukálunk BASIC-ben, hasonlóan egyszerűen:

3. PROGRAM – SPIRÁL

```
50 INPUT NO,SZ
100 GRAPHIC 1,1
110 LOCATE 160,90
120 FOR I=1 TO 90
130 DRAW TO R;S
140 R=R+NO:S=S+SZ
150 NEXT I
```

NO=3:Sz=121 paraméterekkel ld. μ Magazin 85/1, 5/a ábra ill. hasonlót BIT-LET 1984/április, 2. ábra.

4. PROGRAM – CSILLAG

```
50 INPUT S1
100 GRAPHIC 1,1
110 LOCATE 100,120
120 FOR I=1 TO 10
130 DRAW 1 TO 100;SZ
140 SZ=SZ+S1
150 NEXT I
```

S1=144 mellett ld. BIT-LET 1984/aug. 2. ábra

Előző (CSILLAG) programot picit megváltoztatva:

5. PROGRAM

```
130 DRAW 1 TO 50;SZ
135 SZ=SZ+288:DRAW TO 50;SZ
```

szintén valami „csillagszerű" adódik. Sajnos a μM 85/1. számból hiányzik a 4. ábrához tartozó lista, így azt csak hozzávetőlegesen lehet lefordítani.

6. PROGRAM

```
50 INPUT OL,FO
100 GRAPHIC 1,1
120 FOR I=0 TO OL
125 LOCATE 100,100:LOCATE
+50;I*360/OL:FOR J=1 TO FO
127 FOR K=1 TO 3
130 DRAW 1 TO 50;SZ
135 SZ=SZ+120:NEXT K
140 SZ=J*360/FO:NEXT J
150 NEXT I
```

OL=5:FO=8 paraméterekkel az ott láthatókhoz hasonlót kapunk.

A cikk 6. ábráját generálja az alábbi program NO=7-tel:

7. PROGRAM – ZÁRT SPIRÁL

```
50 INPUT NO
100 GRAPHIC 1,1
110 LOCATE 160,100
115 L=5:SZ=0
120 FOR I=0 TO 1000
130 DRAW TO L;S
135 SZ=SZ+NO+360*(360<SZ)
140 S=S+SZ+360*(360<S)
150 NEXT I
```

Az eddigi igen egyszerű példácskák jobb megértését szolgálja, ha megpróbálkozunk további „INPUT értékekkel". Természetesen nem érdemes akármit beírni, először kísérletezzünk, aztán talán valami „szabályosságra" is fény derül.

Az „oszthatóság, relatív prímek, legnagyobb közös osztó", stb. fogalmak sokat segítenek!

Mit lehet mindebből tanulni?

Egy újabb „ismert" LOGO látványt rajzol az alábbi pár sor:

8. PROGRAM

```
50 INPUT NO,SZ,OL,FO
100 GRAPHIC 1,1
110 LOCATE 160,100
120 FOR I=1 TO OL
130 R=NO:FOR J=1 TO FO
140 DRAW TO R;S
150 R=R+NO:S=S+SZ
160 NEXT J,I
```

Ennél a programnál érdemes egy kis időt szentelni a paramétereknek! Az ábra durván négy „részből" áll, erre triviálisan utal az OL=4 érték. Kevésbé nyilvánvalóan, de sokkal határozottabban utal SZ=90! Ez a két érték szinte determinisztikusan összetartozik. NO=3 az ábra „szellősségét", FO=17 pedig a „bonyolultságát" befolyásolja. A nem túl fontos NO paramétert egy kis próbálgatással könnyű beállítani. Próbálkozzunk viszont FO=16-tal! Ez bizony sokat változtat. És FO=14? Vegyük észre, hogy OL és FO értékek legnagyobb közös osztója alapvetően meghatározza az ábra lehetőségeit.

Próbáljunk kilépni a négyzetek világából! CSILLAG programban 144 volt a bűvös szám, próbálkozzunk most is a 7, 144, 5, 16 sorozattal! Kicsit zsúfolt az ábra, amin FO=17 sokat segít. Már csak a misztikus 144-es számot kellene megfejteni, amihez vegyük észre, hogy 144=180–180/5.

Ezek után az az érzésünk, hogy már minden rendben, pláne ha kipróbáljuk az 5,120=180–180/3,3,11 paramétereket is. Elbizakodottság ellen jót tesz az 5,135= =180–180/4,4,11 sorozat. Bizony már a csillagnál is feltűnhetett, hogy a páros és páratlan oldalú sokszögek képzése eltérő. OL=páratlan esetén az adott képlet tökéletes. Tapasztalati úton OL=4-hez SZ=90, OL=8-hoz SZ135 tartozik. OL=6-hoz sajnos nincs megfelelő szög, VAJON MIÉRT? (Hogyan tudunk hatágú csillagot rajzolni?)

Sajnos 4, 160, 160, 9, 23 sorozat eltér az eddigiektől „sorrendben". Ugyanakkor a 2, 72, 5, 21 is „jól néz ki", sőt az esetek kimeríthetetlenségét jelzi, ha beütjük az 5, 123, 6, 20 sorozatot.

Búcsúzóul egy elég sokoldalú DEMO-programot közlünk, aminek paraméterei „irányítottan véletlenek". Különböző sokszögekkel „parkettázunk", és minden további szövegnél többet ér a program tanulmányozása, sőt továbbfejlesztése.

9. PROGRAM

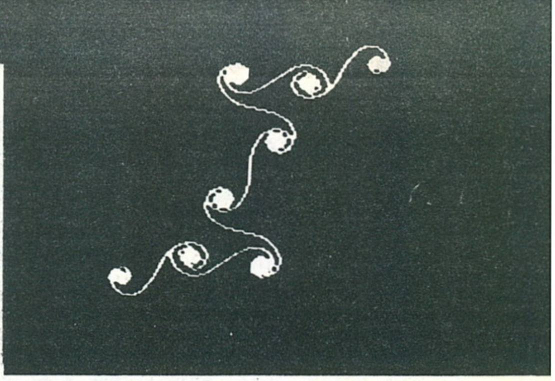
```
20 DEFFNR(X)=INT(RND(1)*X+1)
29 FOR I=5 TO 9 STEP 2:READ D1(I):NEXT I
30 FOR I=3 TO 10: READ D(I):NEXT I
31 DATA  .4 , .5, .6,1.5,1.5,1.5 ,1.73,1.5 ,1.85,1.5,1.9
200 GRAPHIC 1,1
210 M=FNR(8)+2:SZ=360/M:RR=INT(8+2*M/3+FNR(25/M))
212 IF ABS(M-5)=1 AND RND(1)>.3 THEN S1=0:GO TO 220
215 S1=(FNR(6)*1)*15
220 FF=0:IF M/2<>INT(M/2) THEN FF=SZ/2
225 CIRCLE 1,160,100,RR,,,S1,SZ:PAINT 1,160,100
230 R1=0:N=M:K=1
240 IF M=3 OR FF=0 THEN GOSUB 300 :ELSE GOSUB 400
250 K=K+1:IF R1<150 THEN 240
270 GETKEY A$
299 GO TO 200
300 FOR I=0 TO N:LOCATE 160,100
305 R1=RR*(K*D(M)+SGN(FF)*(K/2-INT(K/2)))
310 LOCATE +R1;S1+(I+.5)*SZ+FF*(K*2-1)
320 IF FF=0 AND K=1 THEN CIRCLE1,,RR,,,S1,SZ:ELSE X(I)=RDOT(0):Y(I)=RDOT(1)
330 NEXT I
340 IF FF=0 AND K=1 THEN 390
350 L=K:IF M=3  THEN L=K+INT((K+1)/2)
370 FOR I=1 TO N:FOR J=0 TO L
375 X1=(J*X(I-1)+(L-J)*X(I))/L:Y1=(J*Y(I-1)+(L-J)*Y(I))/L
380 CIRCLE 1,X1,Y1,RR,,,FF*K+S1,SZ:IF K/2=INT(K/2) THEN PAINT 1,X1,Y1
385 NEXT J,I
390 RETURN
400 R1=R1+RR*D(M):IF K/2=INT(K/2) THEN R1=R1+D1(M)*RR:ELSE R1=R1+RR/2
405 FOR I=0 TO N:LOCATE 160,100
410 LOCATE +R1;S1+(I+.5)*SZ+FF*(K*2-1)
420 X(I)=RDOT(0):Y(I)=RDOT(1)
430 NEXT I
450 L=K+INT((K+1)/2)+(K>2)
470 FOR I=1 TO N:FOR J=0 TO L
475 X1=(J*X(I-1)+(L-J)*X(I))/L:Y1=(J*Y(I-1)+(L-J)*Y(I))/L
480 CIRCLE 1,X1,Y1,RR,,,FF*K+S1,SZ:IF K/2=INT(K/2) THEN PAINT 1,X1,Y1
485 NEXT J,I
490 RETURN
```

Utószó

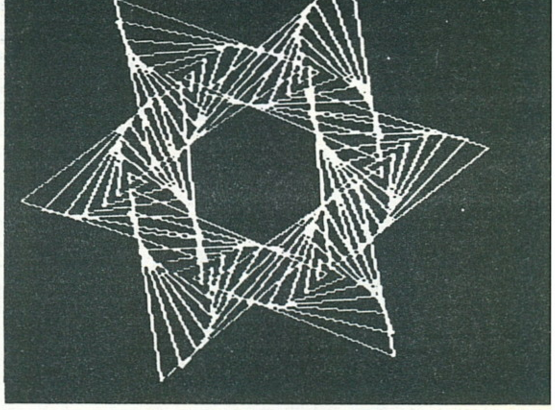
Véletlenül sem az volt a célom, hogy a LOGO fölé emeljem a BASIC nyelvet. Csupán egy igen vonzó lehetőséget kívántam vázolni, egyrészt mindenki szórakoztatására, másrészt ínyenceknek továbbgondolásra. Meggyőződésem ugyanis, hogy ha ez a szemléletmód is beépül a közoktatásba, akkor sokkal hatékonyabbá teheti a középiskola sok ponton enervált geometria tanítását! Ugyanakkor az éber olvasó észrevehette, hogy Ury László cikkének 7. és 8. ábrája diszkréten hiányzik a jelen gyűjteményből. A „rekurzív eljárás-hívás" ugyanis hiányzik a BASIC-ből, sőt, sajnos azt mondhatjuk, hogy „teljesen idegen a szellemétől". Egyszerűbb esetei (önmagát hívó szubrutin) megvalósíthatók, de mindenképpen különös elővigyázattal kezelendő. Ez a témakör feltétlenül érdemelne egy alaposabb cikket.

T. T.

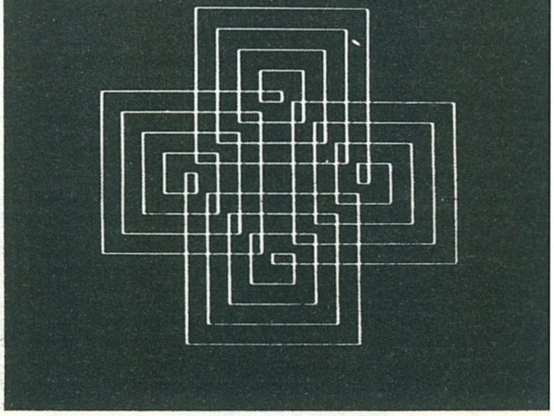
7. PROGRAM



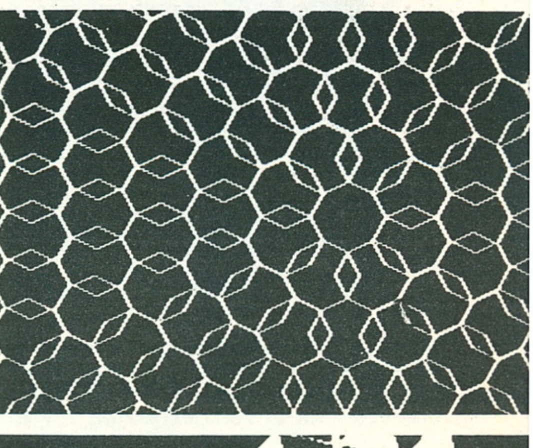
8. PROGRAM (5, 123, 6, 20)



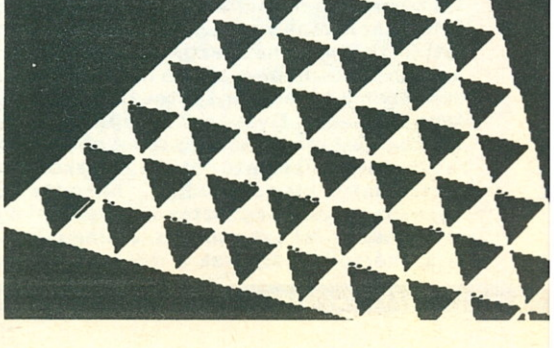
8. PROGRAM (3, 90, 4, 17)



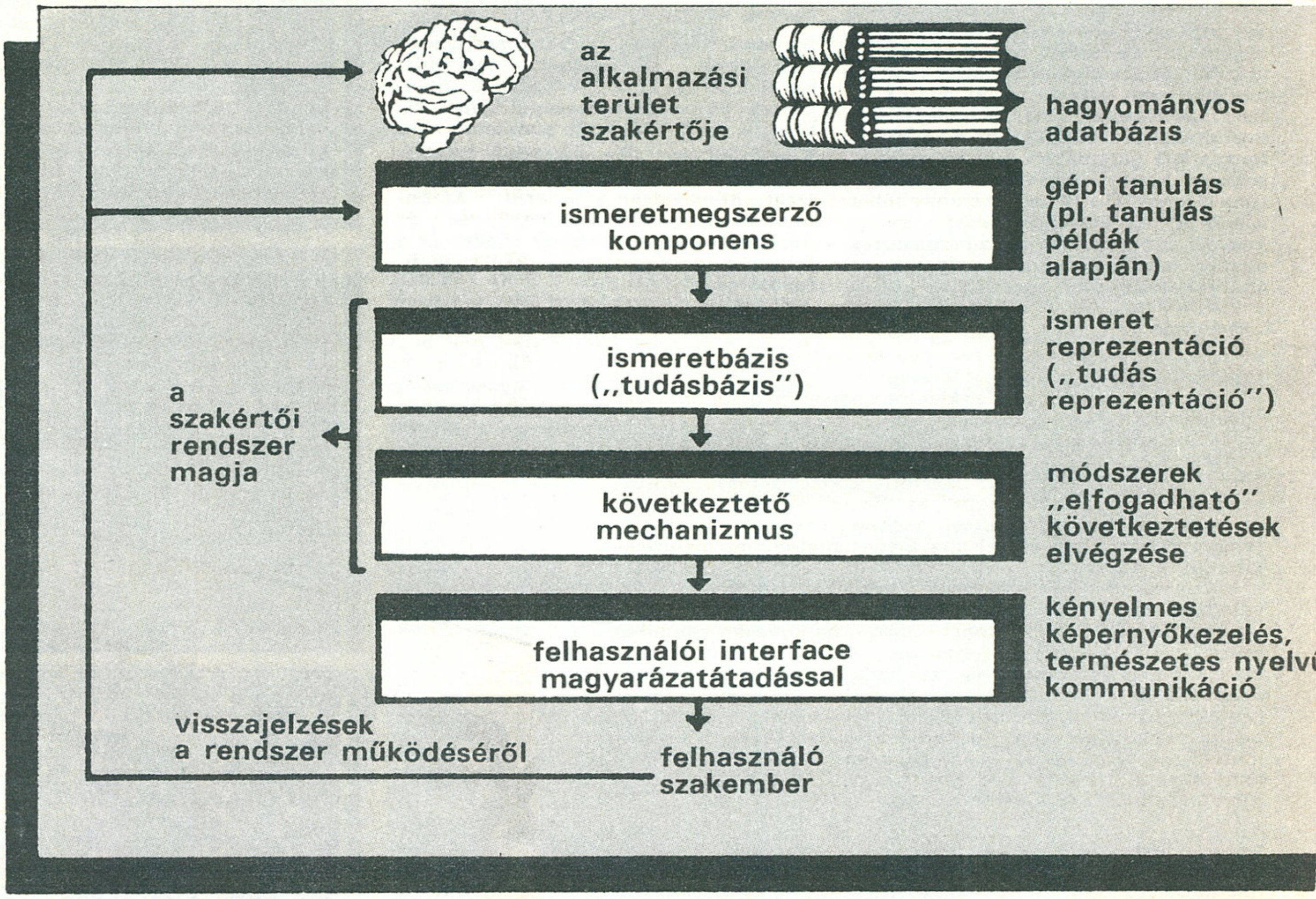
9. PROGRAM



9. PROGRAM



Néhány szó szakértői rendszerekről



SZAKÉRTŐI RENDSZEREK LEEGYSZERŰSÍTETT SÉMÁJA

Legutóbbi ígéretünket csak részben tartjuk be. Úgy ígértük ugyanis, hogy a 2 Gépnyerő pályázatra érkezett szakértői rendszert modellező programok egyikét is közöljük majd. Sajnos azonban kiderült, hogy a jól működő programok olyan programozástechnikai hibákat, csúnyaságokat tartalmaznak, amelyek miatt nem érdemes közölni egyiket sem. A szakértői rendszerekről ígért alapozó cikk viszont elkészült, így ezt a részét teljesítjük az ígéretnek.

A szakértői rendszerek olyan számítógépes rendszerek, amelyek egy szűk alkalmazási terület szakembereit segítik oly módon, hogy a terület elismert szakértőitől megszerzett ismeretek felhasználásával „intelligens" következtetések levonására képesek. Összevetve a numerikus adatokkal dolgozó algoritmikus programokkal, a szakértői rendszerekben az ismeretanyag szimbolikus információként van tárolva, és feldolgozása nem lépésenként előírt algoritmusokkal, hanem a szimbolikus logika és a gyakorlati életből vett heurisztikus módszerek felhasználásával történik. Nemcsak ebben, „gondolkodásmódjukban" hasonlítanak az emberhez, hanem abban is, hogy az általuk tárolt ismeretek, „tudásuk" lehet hiányos, téves, olykor ellentmondásos. Ezért a rendszerek tévedhetnek, továbbá saját problémamegoldási tapasztalataik alapján korrigálhatják, bővíthetik ismereteiket, azaz tanulhatnak is. Gyakorlati jelentőségük abban van, hogy az adott alkalmazási területen működő szakember (azaz a rendszer felhasználója) mindennapos tevékenységéből átvállalják magukra az emberi intuíciót, mérlegelést, döntéshozatalt igénylő munkálatok mechanizálható részét. A szakember így nem fárad úgy el a nagy odafigyelést igénylő, egyébként monoton és unalmas rutinfeladatok megoldásában – helyette azokat elvégzi az ilyen feladatokat hatékonyabban és objektívabban elvégezni képes gép. Előnyös munkamegosztás tud így kialakulni az ember és a gép között magasszintű szakértelmet kívánó tevékenységek során is: mindkét fél azt csinálja, amiben ő a „jobb".

A szakértői rendszerek a mintegy 30 éve beindult mesterséges intelligenciakutatások során kifejlesztett problémamegoldó

technikákat egyesítik a korszerű szoftver technológia eszközeivel. Ellentétben a hagyományos számítógépes rendszerekkel, ezek általában nem egzakt eredményt adnak, hanem (tárolt ismereteikhez mérten elfogadható!) tanácsot, javaslato(ka)t. Sok rendszer például explicit módon kezeli a tárolt ismereteket – és a felhasználásukkal kihozott következmények – bizonytalansági mértékét. Természetes igény ezek után, hogy tanácsaikat, javaslataikat – a felhasználó kérésére – meg is tudják indokolni. E magyarázatadási képességet elvárja a felhasználó a rendszer működése során lezajló párbeszédnél is olymódon, hogy – megintcsak a felhasználó kívánságára – érthető módon meg tudják indokolni a felhasználó számára feltett kérdéseiket, megtett következtetési lépéseiket.

Egy ilyen „mesterséges" szakértő előnye az „emberi" szakértővel szemben az, hogy ismereteit, „tudását" nem felejti el (az „halhatatlan"), és azt következetesen mindig ugyanúgy használja a problémamegoldás során. E tárolt ismereteket könnyű dokumentálni és több példányban előállítani, terjeszteni. A szakértői rendszerek felhasználásával hosszú távon olcsóbban juthatunk egy (vagy akár több) elismert szakértő ismeretein alapuló tanácshoz, javaslathoz, mintha minden esetben kifizetnénk a szakértési díjat (és még várnunk is kell a szakértő megérkezéséig esetleg napokat).

Vannak azonban hátrányai is a szakértői rendszereknek az „emberi" szakértővel szemben, amikről nem szabad elfeledkeznünk. Az ember kreatív változó körülményekhez alkalmazkodó, környezetével különböző érzékszervei révén sokféle kapcsolatot tartó, józan ésszel rendelkező, tanulni többféle módon képes élő „rendszer". Nem így az ihlettelen, lélektelen, „saját" józan ésszel nem rendelkező szakértői rendszer, amely környezetével általában csak egyféle kapcsolatot tart, tisztán szimbólumokban gondolkodik, beépített látószöggel és csak technikai tudással valamint jelentős, még korlátozott tanulási képességgel rendelkezik. (A szakértői rendszerek továbbfejlesztésében a következő évek egyik feladata épp a gépi tanulás technikáinak kidolgozása.)

Az előző oldalon levő ábrán egy szakértői rendszer „szereplői" láthatók. Nincs feltüntetve, bár igen fontos emberi résztvevő egy ilyen rendszer kiépítésénél a szakértői rendszer specialistája, aki egyébként ma szűk keresztmetszete a szakértői rendszerek készítésének. E specialista a rendszer kifejlesztése során az alkalmazási terület szakértőjével folytatott alapos konzultációk során „kinyeri" annak tudását (pontosabban: annak „elmondható" és formalizálható részét), és az alkalmazott következtetési mechanizmust – be tudja tölteni a rendszer ismeretbázisába. E specialista dolga még felépíteni a rendszer többi komponensét is. Kiemeljük ezek közül a szakismeretek megszerzését, azoknak az ismeretbázisba való betöltését, ottani módosítását, valamint konzisztenciavizsgálatát végző komponenst, melyet, ha ügyesen van elkészítve, maga a szakértő is használhat. S ez elemi követelmény is, ugyanis egy szakértői rendszer soha sincs teljesen készen: a szakértő az időközben tudomására jutott ismeretekkel gyakran aktualizálja, a felhasználó visszajelzéseinek megfelelően pedig indokolt esetben módosítja az ismeretbázist. Kényelmetlen (és értelmetlen) lenne e karbantartási jellegű tevékenységre a már más rendszerrel foglalkozó szakértői rendszerspecialistát felkérnie.

Az ábrán látható módon jól el van különítve a tárolt szakértői tudás, az ismeretbázis az azt „motorként meghajtó" következtetési mechanizmustól. Ez elemi szervezési követelmény az ún. ismeretbázisú rendszereknél, amelyet nemcsak az ismeretbázis előbb említett gyakori aktualizálása, hanem a végrehajtó mechanizmus több rendszerben való felhasználásának igénye is megkívánt.

Az első szakértői rendszerek az 1970–80-as években léptek ki az USA mesterséges intelligenciakutatással foglalkozó laboratóriumaiból. Nyilvánvalóvá vált eddigre a kutatók számára, hogy egy számítógépes program „problémamegoldó ereje", így intelligenciája is lényegében attól függ, hogy mennyire magasan kvalifikált és specifikus, mennyire széleskörű és mély, végül mennyire nagyméretű az általa kezelt ismeretanyag. Az első, immár klasszikusnak számító szakértői rendszerek: a DENDRAL (1971); szerves vegyületek topológiai struktúráját meghatározó interpreter, a MYCIN (1976); a vér bakteriális fertőzéseinek diagnosztizálására, valamint antibiotikumos kezelésére javaslatot tevő rendszer, a PROSPECTOR (1978); egy terület ásványlelőhelyeinek felkutatásában segítséget adó tanácsadó rendszer. Jellemző mai alkalmazási területek: kémia, számítógépes rendszerek, elektronika, mérnöki alkalmazások, geológia, orvosi alkalmazások és – természetesen – katonai alkalmazások.

A szakértői rendszerek kezdetben kizárólag LISP-ben készültek (az USA-ban még ma is ezt a leggyakoribb.) Az 1970-es évek végén már készültek PROLOG-ban is ilyen rendszerek. Az SZKI-ban kifejlesztett MPROLOG-ban készült például a szekszárdi kórház számára a SZÁMALK-ban és az SZKI-ban készített diagnosztizáló rendszer. Vannak már PASCAL, FORTRAN, sőt C és assembly nyelven írott szakértői rendszerek is, azonban ezekben a következtetéseknél igényelt szimbólum-manipulációkat és az ún. „visszalépéses keresést" keményen be kell programozni. A tömegtermelés beindulásával természetesen megjelentek a szakértői rendszerek kifejlesztését támogató rendszerek, eszközkészletek is. Sőt, a piacon már kaphatók mikrogépeken futtatható szakértői rendszerek és – fejlesztő rendszerek. Utóbbiakra néhány példa: APES, ESP Advisor, Expert Easy, Insight 1 és 2, Savoir, Xsys. Az Insight 1 például mindössze 95 dollárba kerül, és elegendő egy 128 K operatív tárú IBM PC a futtatásához. A tárolható tudás kicsiny volta miatt természetesen igen leszűkítetten alkalmazható.

Összefoglalásképpen próbáljunk meg választ adni a gyakran elhangzó kérdésre, hogy valódi szakértők-e a mai szakértői rendszerek, illetve hogy képesek-e helyettesíteni, nélkülözni az embert? A válaszok: nem, mert amikor egy feladatra nem tudnak elfogadható megoldást adni, nem képesek a szakterület tudásanyagának mélyebben fekvő szintjein felvetni újra a problémát (mint az ember). Nem, mert tudásuk betű szerinti, váratlan szituációkban kiderül, nincs „józan eszük". Nem, mert (egyenlőre még) nem tudnak saját tapasztalataikból (igazán) tanulni. Az utóbbi probléma enyhítése, részbeni megoldása irányába erőteljes kutatások folynak. A Tudomány című folyóiratban hamarosan megjelenik D. Lenat egy cikke magyar fordításban, ahol példákat mutat be EURISKO nevű, indukció útján tanuló rendszer eddigi sikeres kísérleti alkalmazásáról. A tanuláson kívül még sok kutatási, fejlesztési feladat van az ismeretek reprezentálása és felhasználása, az inegzakt következtetés, a magyarázatátadás, az ismeretek megszerzése és az ismeretbázis validálása témákban. Nyugaton 1988-ra a szoftverek 2 %-át a szakértői rendszerekben realizált mesterséges intelligenciatechnikák fogják alkotni. Mielőbb meg kell tehát ismerkednünk ezekkel, és mielőbb napi gyakorlattá kell tennünk ilyen – vagy az eddigieknél legalább kicsit „intelligensebb" – szoftverek készítését.

Sántáné Tóth Edit

ötlet

SZOFTVER ÖTLETEK



VC 20-ON BOTKORMÁNY ÁLLAPOT LEKÉRDEZÉSE

A program kihasználja a megszakítórutin átcímezhetőségét, ezért futtatása után a BASIC programok törölhetők és az SYS-sel meghívott gépi kódú program tovább működik úgy, hogy közben a gépen rendesen dolgozhatunk. Megállítani csak STOP–RESTORE-rel vagy a gép kikapcsolásával tudjuk. STOP–RESTORE esetén SYS673-al újraindítható.

A botkormány állapota az 1000-es címről lekérdezhető, így nem szükséges a VIA #chip átcímzése, ami könnyen a gép lemerevedését okozhatja.

```
10 FOR A=673 TO 762:IFA=697 THEN A=724
20 READ B:POKE A,B:NEXT
30 SYS 673
40 DATA 120,173,20,3,141,251,2,173,21,3,141,252,2,169,212,141,
   20,3,169,2,141,21,3,96
50 DATA 162,255,142,35,145,160,0,140,34,145,140,33,145,173,32,
   145,41,128,141,232,3
60 DATA 173,17,145,41,60,13,232,3,141,232,3,140,35,145,142,34,
   145,76
```

Juhász György, 3100 Salgótarján. Pécskő u. 3.

Kedves Szerkesztőség!

Küldök Spectrumra egy néhány olyan POKE címet és értéket, amely hasznos lehet:

cím	érték	
23617	1	E–kurzor
23617	2	G–kurzor
23617	4	K–kurzor
23617	0	esetén:
23658	0	L–kurzor
23658	8	C–kurzor

Ezen INPUT utasításkor lehetnek hasznosak, a programleállítás utáni kurzorra csak a 23658 címen levő értéknek van hatása.

Monori Zsolt 3580 Leninváros, Malinovszkij u. 43.

Az egyik, nemrégiben a Posta rovatban megjelent válaszunkhoz érkezett egy kiegészítés. A sportszerűség azt kívánja, hogy ezt is közzétegyük. Tessék!

A BIT-LET november 20-i számában olvastam Lajos Sándor levelét, amiben a ZX 81 billentyűvizsgálatáról ír. Tulajdonképpen csak a lap által feltett kérdés miatt fogtam tollat. A kérdés: Egyszerűbb-e a

```
10 IF PEEK 16421=223 AND PEEK 16422=253 THEN GOTO10,
```

mint a

```
10 IF INKEY$="P"THEN GOTO 10
```

programsor.

A válasz egyértelműnek tűnik. Nem, sőt amellett, hogy nem egyszerűbb, még lassúbb is az első sor. De érdemes a kérdéssel mélyebben foglalkozni. Ha valaki kétszemélyes játékot szeretne írni, vagy egyéb okból kettő vagy esetleg több billentyű állapotát egyidőben vizsgálni, nem sok hasznát veszi az INKEY$-nak, hiszen ez egy olyan függvény, aminek az értéke egyetlen karakter. A mintaprogramban vizsgált két byte-nak, pontosabban 16 bitnek itt lehet nagy szerepe.

Az említett számban megjelent táblázatot most egy kicsit kibővítve írom le, és itt nem is a vizsgált byte-ok értéke, hanem az adott byte egy billentyű lenyomásának hatására 0-ba álló bitjének helye lényeges.

Könnyen látható, hogy egy-egy billentyű lenyomásának hatására az említett 16-bitből 2 db változik 0-ra, a többi nem változik meg. Az 16421 bitjei a billentyűzet félsorait mutatják, az 16422 bitjei pedig az adott félsorban levő billentyűket, így mintegy rácsot létrehozva, ahol a rácspontokban egy-egy billentyű található. Az igazi előnye persze abban van, hogy a rácspontokat külön-külön tudjuk vizsgálni, így bármely billentyű állapotáról pontos információt nyerhetünk, a többi billentyűtől függetlenül.

A vizsgálat módja:

```
IF 2*(PEEK 16421/2**(t2+1)–INT(PEEK 16421/2**
(t2+1)))<1AND 2*(PEEK 16422/2**(t1+1)—INT(PEEK
16422/2**(t1+1)))<1 THEN
```

A t1 illetve a t2 értékeit a mellékelt táblázatból lehet kimásolni.

Ezzel túljutottunk az első nehézségen, amit az INKEY$ szegényes szolgáltatásai teremtettek, de szembe kell néznünk azzal, hogy ezeknek a feltételeknek a kiértékelése komoly időbe telik, és az amúgy sem villámgyors BASIC-programokat tovább lassítja. Egyetlen valamire való megoldás, a gépi kód használata.

Leírnak egy rutint, ami egy adott billentyű állapotától függően F1, illetve F2 címre adja át a vezérlést.

```
LD HL, 16421
BIT t2, (HL)
JP NZ, F2
INC HL
BIT t1, (HL)
JP NZ, F2
F1:
F2:
```

Itt a t1 illetve a t2 ugyanaz, mint a BASIC sorban, azaz a táblázatból lehet kimásolni.

Az JP NZ helyére esetleg JR NZ-t lehet írni, ha az ugráshatárok ezt megengedik. Remélem sokaknak segítettem abban, hogy megírhassák a megálmodott programjaikat.

Kovács Gábor Vác, Lemez u. 7.

	1	2	3	4	5	6	7	8	9	0	
t1	1	2	3	4	5	5	4	3	2	1	
	253	251	247	239	223	223	239	247	251	253	PEEK 16422
	3	3	3	3	3	4	4	4	4	4	
t2	247	247	247	247	247	239	239	239	239	239	PEEK 16421
	Q	W	E	R	T	Y	U	I	O	P	
t1	1	2	3	4	5	5	4	3	2	1	
	253	251	247	239	223	223	239	247	251	253	PEEK 16422
t2	2	2	2	2	2	5	5	5	5	5	
	251	251	251	251	251	223	223	223	223	223	PEEK 16421
	A	S	D	F	G	H	J	K	L	ENTER	
t1	1	2	3	4	5	5	4	3	2	1	
	253	251	247	239	223	223	239	247	251	253	PEEK 16422
t2	1	1	1	1	1	6	6	6	6	6	
	253	253	253	253	253	191	191	191	191	191	PEEK 16421
	SHIFT	Z	X	C	V	B	N	M		SPACE	
t1	0	2	3	4	5	5	4	3	2	1	
	254	251	247	239	223	223	239	247	251	253	PEEK 16422
t2	–	0	0	0	0	7	7	7	7	–	
	255	254	254	254	254	127	127	127	127	255	PEEK 16421

	7.	6.	5.	4.	3.	2.	1.	0.
251=	1	1	1	1	1	0	1	1
223=	1	1	0	1	1	1	1	1
127=	0	1	1	1	1	1	1	1
239=	1	1	1	0	1	1	1	1

BIT-LET POSTA

BP. 1986

Körülbelül 1 éve használok HT–1080Z típusú számítógépet és körülbelül egy negyedéve a gépi kódú részét is használom. Most azt szeretném Önöktől, ha a BIT-LET-ben vagy levélben szereznének nekem
1. SCROLL programot (lehetőleg minden irányban)
2. Egy olyan ASSEMBLER programot, ami ha a nyilakat lenyomom, akkor a program különböző részére ugorna.
3. Múltkori (23). számukban láttam egy SPRITE programot, ami azonban gépi kódban volt. Ha lehetséges, akkor ezt a programot is írják meg ASSEMBLER-ben és közöljék.
Ezekből a programrészekből egyébként szeretnék majd egy komolyabb programot összeállítani. Előre is köszönöm segítségüket!
***Fáczán László**, 8800 Nagykanizsa, Kazanlak 7/B*

Tisztelt Fáczán László!
Már többször megírtuk, hogy programokat szerezni, küldeni és átírni nem tudunk. Levelét közzétesszük, hátha valaki segít. Azonban engedjen meg két megjegyzést:
1. (SCROLL program). Ha komolyabb programok írásával foglalkozik, akkor ennek megírása nem okozhat gondot.
2. (SPRITE) A gépi kódú programot egyszerűen fordítsa vissza egy DISASSEMBLER-rel. (Megértése nehéz lesz!)

Nekem C 116-os számítógépem van, amely ugyanaz, mint a 16-os, csak a doboza más. Véleményem szerint is igen jók ezek a komputerek, csak az a problémám, hogy a gépi kódú programozásról sajnos nem tudok információkat szerezni. Ezért szeretném megkérdezni, van-e olyan könyv, ami a C 116-os gépi kódú programozásához felhasználható? Másik kérdésem: van-e a C 16-oson, illetve a 116-oson SPRITE, és ha igen, hol, melyik kezdőcímen?
***Joanelli Tamás**, 8000 Székesfehérvár, Szepesi u. 5.*

A második kérdésre egyszerű a válasz, NINCS. A C 116-ra gépi kódú programozásról könyv nemigen jelent meg, de maga a gépi nyelv megegyezik a C 64-gyel.

Még a nyáron kaptam ZX Spectrum + típusú számítógépemet egy Interface 1-el, és egy microdrive-al. Ez utóbbiakhoz kapcsolódik két kérdésem.
1. A gép csak a gyári Demo cartidge-ról tud programot beolvasni. (írni erre sem tud)
Teljesen új kazettát próbáltam formattálni „Cart. 1" névre. Az eredmény ez volt: (CAT 1): Cart.1–0
*A SAVE * "m"; 1; "prog 1" eredménye pedig a Microdrive Full hibaüzenet volt.*
Kérem írják meg a hiba okát, és elhárításának módját.
2. Melyek azok a kapuk, amelyek a microdrive ki- és bemenetét vezérlik, és mi az egyes bitek funkciója?
***Sovány László**, 7400 Kaposvár, Géza utca 1.*

Problémájával megkerestük a MICROTEAM GM-et, akik szervizeléssel foglalkoznak.
1. Sajnos a leírás alapján valószínűleg a microdrive-ban levő ULA a hibás, amit beszerezni nem lehet.
2. A vezérlőjelek ismertetésének nincs értelme, ugyanis olyan áramkört, amely ezt felhasználja, házilag készíteni nem lehet.

Tisztelt Szerkesztőség!
Régóta olvasom a BIT-LET-et, és azért fordulok Önökhöz, mert véleményem szerint ez a legjobb szaklap. (Olaj... – A Szerk.)
A problémám a következő: ha nekem lenne egy C 64-em és egy 1660-os modemem, ezzel tudnék-e valamit kezdeni? Arra gondolok, hogy a telefonvonal sose, vagy csak esetenként használható. Elképzelhető-e Magyarországon valamilyen partner, akivel kommunikálhatok?
Előre is köszönöm
***Török Ágnes**, 8800 Nagykanizsa, Zrínyi u. 13.*

Nem tudunk ilyen, már üzemelő rendszerről, de mi magunk is szívesen vennénk, hogy ha valakinek van, megosztaná velünk tapasztalatait. (Szívesen részt vennénk esetleges próbakapcsolat létrehozásában is!)

„Tisztelt Szerkesztőség! A Bit-let októberi számában cikk jelent meg Kispál István szerzőtől ‚Ékezetes betűk nyomtatása' címmel. Ugyanez a cikk megjelent a Mikroszámítógép Magazin 1985/5. számában is. A Bit-let novemberi számában a szerkesztő nyílt levelet intézett a szerzőhöz, amelyben – joggal – elmarasztalta.
Szíves tudomásukra hozom, hogy a sajnálatos eset a Mikroszámítógép Magazin szerkesztőségében uralkodó, eléggé el nem ítélhető rendetlenség miatt következett be, ugyanis a szerző május 9-én nekünk megküldött cikkének közlésére vonatkozóan nem nyilatkoztunk, s azt – minden külön értesítés helyett – leközöltük. Tisztelő hívük: Könyves Tóth Pál, a Mikroszámítógép Magazin (mindenért) felelős szerkesztője".

Köszönjük az észrevételt. Őszintén szólva a közöltektől nem döltünk romjainkba. Nem kérünk elnézést Tóth Páltól. Úgy gondoljuk ugyanis, hogy a szerző mindaddig, amíg lemondó levelet nem kapott a magazin szerkesztőségétől, etikai vétséget követett el a kézirat hozzánk való elküldésével. (Zárójelben jegyzem meg, hogy mi sem küldünk értesítést szerzőinknek a beküldött anyagok sorsáról, csak ha erre vonatkozó külön kérést tartalmaz a beküldött anyag.)

HIBAIGAZÍTÁS!!

Sajnos az ördög bújt belénk a múlt hónapban. Még ennyi hiba egy számunkba soha nem került egyszerre. Két nagyobb programot közöltünk, mindkettőt súlyos hibákkal.
A Spectrumra közölt rendezőprogramból kiirtandók a 41, 42, 430, 450-es számú sorok.
A javítandó sorok pedig:

```
10-BEN:..3. UJ REKORD FELVITELE
110-BEN:64899 HELYETT 64956
150-BEN:65040 HELYETT 65097
620-BAN:A$(K,20) HELYETT A$(K,E)
750-BEN:65073 HELYETT 65124
800-BAN:65069 HELYETT 65120
1030-BAN:64899 HELYETT 64957
  A KOVETKEZO 4 SORRAL MODOSUL
400 INPUT#N;(A(E)+E);TAB 5;
 SZERZO.. CIME... S T OLV. "
 IRD-> ;LINE K$:IF K$= * THEN
GOTO T
410 IF LEN K$<>A(T)-E THEN
GOTO 400
420 LET A(E)=A(E)+E:LET A$(A(E))
=   +K$:GOTO 800
550 LET A$(KK,K+E  TO K+LEN K$)
=K$:GOTO T
AZ ASSEMBLY LISTA 62. SORA
  LD A,(HL)
```

A C 16-ra közölt karakter tervező program listázásánál súlyos mulasztást követtünk el. Ugyanis a C 16 helyett C 64-be töltöttük be a programot, s így printeltük. (Tetszik tudni nálunk is úgy van ez, hogy ahol van C 16, ott éppen nincs printer, ahol meg van egy használható printer végre, ott nincs ott a C 16.) Így történhetett meg, hogy a listában olyan marhaságok olvashatók, amilyenek. Ezek szerint minden LIST utasítás helyett COLOR-t kell írni, minden SAVE helyett CHAR-t. Az 520-as és az 1370-es sorban DIM helyett HEX$ a helyes, a 260-as és 1390-es sorban pedig GOTO helyett ELSE szükséges.
Minden kedves programozó barátunktól súlyos elnézést kérünk, s kérjük, aki olvassa ezeket a sorokat, szóljon azoknak, akikről úgy gondolja, hogy megüti a guta, hogy a gépbe írott programja nem működik!

ZSÁKBAMACSKA NYERŐ ÉRTÉKELÉSE

Pályázatunkra csak 11 megfejtés érkezett, ami nem sok. Ráadásul a 11 megfejtő közül csak kettőnek sikerült a feladat állítását maradéktalanul bizonyítani, Kurusa Árpádnak (Szeged, Alföldi u. 21. 6725) és Szalkai Istvánnak (Bp. Budaörsi út 95–101. A épület 925. szoba, 1118). Szalkai István bizonyítása a miénkhez hasonló (itt jegyezzük meg, hogy sajnos a megoldás közlésénél becsúszott egy sajtóhiba), természetesen 3x3-as négyzetekre kell felosztani a játékmezőt. Kurusa Árpád bizonyítása kicsit egyszerűbb, ezért (s mivel a feladat ilyen nehéznek bizonyult) röviden ezt is közöljük:
Osszuk fel a játékmezőt 2x2-es négyzetre (itt tényleg 2x2-es!), s a kapott négyzeteket gondolatban színezzük feketére és fehérre sakktáblaszerűen! Második mindig ugyanabba a 2x2-es négyzetbe rakja jelét, mint első, s a fekete négyzetekben azt akadályozza meg, hogy elsőnek 2 jele legyen egymás alatt, a fehérekben azt, hogy elsőnek 2 jele legyen egymás mellett. Könnyű látni, hogy ezt mindig megteheti, s ha így játszik, akkor Első nem nyerhet.
A két pályázó között nem sorsolunk, hanem mindketten megkapják a „macskát", azaz a Novotrade ajándékaként 4–4 db Commodore 64 játékkazettát.

Apisz C-16 NYERŐ

3. FELADAT

Biztosan mindenki ismeri a „Vadász és Nyúl" nevű játékot. Egy sakktábla fehér mezőin folyik a játék, kezdetben 4 vadász „áll" a 4 alsó fehér mezőn, s egy nyúl bármelyik felső fehér mezőn. A vadászok csak balra-fel és jobbra-fel léphetnek, a nyúl mind a 4 átlós irányban. Először a nyúl megválasztja a kiindulási mezőjét, aztán a másik játékos lép valamelyik (de csak egy!) vadászával. Ezután a nyúl lép, majd ismét valamelyik vadász, s így tovább.
A vadászokkal játszó játékos nyer, ha be tudja úgy szorítani a nyulat, hogy az már nem tud lépni. A nyúl nyer, ha „kitör" a vadászok közül, azaz hátuk mögé



kerül oly módon, hogy már biztosan nem tudják beszorítani a játék folyamán.
A feladat: írjunk programot, mellyel lehet játszani ezt a játékot. Lehetőség szerint valamilyen módon jelenítse meg a játék állását (pl. karaktergrafikával), figyelje ki nyer, s mind a nyúl, mind a vadász szerepében játsszon jól. A programok legyenek minél rövidebbek, ugyanis a legjobbat közölni szeretnénk.
Többek kérésére közöljük, hogy természetesen a pályázat befejezése után a beküldött kazettákat, disceket visszaküldjük.

C 16

Kérjük levágni és a levélre felragasztani! Beküldési határidő: május 15.

Szerkesztőségünk cime ÖTLET Budapest 1986. Szerkesztő: Angyalosı László. Művészeti szerkesztő: Pribelszky Pál
Tördelő: Domokos Imre. Hírrovat: Csányi György. Vallató: Pogány György. Sorvezető: Székely Jenő. Posta. Halász Péter